Transcopy publishing pictures on the Web now
Transpublishing text on the Web experimentally
TRANSPUBLISHING AND MICROPAYMENTS

To Ted Nelson Home Page

TRANSPUBLISHING AND TRANSCOPYRIGHT

TRANS

TRANSPUBLISHING AND TRANSCOPYRIGHT are a very old idea (from 1960), originally the core of the Xanadu(R) system, which have now been repackaged for the Internet and the World Wide Web.

Here are the two ideas in their present form:

TRANSPUBLISHING means quoting from another document by ARRANGING TO HAVE THAT PIECE DELIVERED FROM THE ORIGINAL.

That means each user receives the quotation from the original publisher, possibly paying the original publisher for it; it means the user has a bridge to the original context. Anything in such a system may be freely re-used and republished (like clip art or boilerplate), since the delivery and purchase of each quoted piece is always from the original.

TRANSPUBLISHING OF PICTURES IS ALREADY POSSIBLE FOR PICTURES ON THE WEB. Why don't we do it a lot? WE DON'T HAVE PERMISSION.

TRANSCOPYRIGHT is the legal system required to make transpublishing possible. It's very simple.

I've chosen the term "transcopyright" as a shorthand way of saying, "I hereby give permission for this material to be used in any new context on the net, as long as it is always delivered from my original (or some other source I authorize)."

And this can be stated very compactly. Ms. Gloriana Brewster may give permission by saying

trans(c) Gloriana Brewster

CENTRAL PIECE

which means the same as

(c) Gloriana Brewster ("I claim the copyright")
AND
"I hereby give permission for this material to be used in any new context on the network."

DETAILED INSTRUX
FOR
PUTTING UP
TRANSPARENCIES

THIS IS A PERMISSION METHOD. I am summarizing this permission in the simple word "transcopyright," which I hope will enter everyday speech, and thus the legal system, in the same way that Bob Wallace's term "shareware" has come to be recognized in everyday speech, and thus the legal system. (Like transcopyright, shareware is also a permission method, summed up in an everyday word. The term "shareware" has legal standing because everyone knows what it means.)

Original article on transcopyright

TRANSPUBLISHING TODAY

1. Pictures on the Web. You can presently transpublish pictures on the Web using the <IMG SRC> tag.

Article on transcopyright for pictures on the Web

Example

2. Text on the Web. At Keio University we have developed a method for text transpublishing on the Web. You may use it through our server. For it to be universally used, however, it will have to become an official part of HTML, accepted by the W3 Consortium.

definition of <TXT SRC> tag

Example of its use (Note:
This brings in a page without bringing in the text quotes.
Click the link at the top of the page
to see the quotes filled in.)

3. Permission formats. We are working on suggested formats for giving the transcopyright permission.

This may seem
clumsy ~~[illegible]~~
~~[illegible]~~
bureaucratic

The copyright
issue
is very big
~~[illegible]~~
[two [illegible] CRUDE
views

~~[illegible]~~

Too

Win-win system
for sharing material

~~People can~~
~~pull~~

~~This is a co~~

Transcopyright is not
a proposal. It works
right now for pictures
and has been approved
by lawyers in the USA
and Japan.

In a world of increasingly
hasty copyright law, this
is intended as a warmheartedly
win-win system for
sharing of materials (and
possibly selling them at
very low prices).

~~Yoshi Tokuji~~

May seem complicated
I see it as simplifying
CLEAR UNDERSTANDING
Answers some of the
objections to copyright

97.01.15

transco.hold.w3

To Ted Nelson's Home Page

Transcopyright is a way of giving permission for virtual inclusion on the Web. It already works for pictures (without payment). Several lawyers in both the United States and Japan have said they like it because it is both legal and clear-- unlike the new copyright laws.

The rightsholder gives permission for the material to be re-used virtually by transclusion in any new context.

The rightsholder is the person who owns rights to the material. This is not necessarily a simple matter, since the rights may include copyright, modelling rights, personality rights, privacy rights, and even trade secret. Ordinarily if you have created a work you are the rightsholder, but if in doubt, consult an intellectual property lawyer.

CAN DO IT NOW

PROPOSED REID FORMATS

Like shareware, foundation can go away

SELL EMBEDDED QUOTATIONS BY THE BYTE — which means small change indeed.

NOT A PROPOSAL BUT A DONE DEAL

P/2 Po, favor
al provide
TEXT
AND PICTURES
UNDER
TRANS©
PERMISSION.

Mail to (a)
for usy doubt
mail to (b)
for source doubt

TEXT TRANSPUBLISHING SITE: Text Quotes Can Come from Anywhere

Transpublication, or transquotation, means arranging to have some parts of an on-line document delivered from somewhere else. This means a quotation is brought to the user's browser from the original. (We may call this "virtual inclusion from elsewhere.")

This can have several benefits to the people involved:
the republisher gets the benefit of freely using that material;
the original publisher, or rightsholder, continues to be the only provider of the material;
the rightsholder benefits from the material getting wider circulation.
Other benefits to the rightsholder that we are working on: arranging for the reader to see easily the original context of the quotation; and automatic payment for that part of the contribution.

EVERYBODY WINS

We believe there will be widespread social benefit from such a system, since everything published under such an arrangement becomes clip art for free re-use, and yet may be bought from its particular publisher.

THE NEED FOR PERMISSION

Such a system, under today's law, must be voluntary. If the rightsholder has not agreed to others' re-use, such republication could be considered a copyright violation-- even if the material is always brought from the original.

Therefore we have worked out a legal system to make it simple to give such permission without giving up any other rights. We call this permission system *transcopyright*. If you publish your document with transcopyright notice, you are giving permission for such virtual republication of its contents-- but you are not giving permission for the material to go on the server of the quoting author. It will go to each user from the original on your server, not from the server where you are quoted.

IT ALREADY WORKS FOR PICTURES

TEXT TRANSPUBLISHING SITE:
Text Quotes Can Come from Anywhere

Transpublication, or transquotation, means arranging to have some parts of an on-line document delivered from somewhere else. This means a quotation is brought to the user's browser from the original. (We may call this "virtual inclusion from elsewhere.")

This can have several benefits to the people involved:
the republisher gets the benefit of freely using that material;
the original publisher, or rightsholder, continues to be the only provider of the material;
the rightsholder benefits from the material getting wider circulation.
Other benefits to the rightsholder that we are working on: arranging for the reader to see easily the original context of the quotation; and automatic payment for that part of the contribution.

EVERYBODY WINS

We believe there will be widespread social benefit from such a system, since everything published under such an arrangement becomes clip art for free re-use, and yet may be bought from its particular publisher.

THE NEED FOR PERMISSION

Such a system, under today's law, must be voluntary. If the rightsholder has not agreed to others' re-use, such republication could be considered a copyright violation-- even if the material is always brought from the original.

Therefore we have worked out a legal system to make it simple to give such permission without giving up any other rights. We call this permission system *transcopyright*. If you publish your document with transcopyright notice, you are giving permission for such virtual republication of its contents-- but you are not giving permission for the material to go on the server of the quoting author. It will go to each user from the original on your server, not from the server where you are quoted.

IT ALREADY WORKS FOR PICTURES

On the World Wide Web, this already works technically for pictures (see our example***). It is easy to transpublish a picture using the <IMG SRC...> tag of HTML. The <IMG SRC...> tag brings in the picture from any other Web site and embeds it in the tag's place. All that's needed is the permission, and we're working to develop good *formats for transcopyright permission*.

THE NEED FOR A NEW TAG

Unfortunately, there is no similar way to transpublish an arbitrary portion of text. We think there is a need for a similar tag for text in HTML, like the picture tag, that will bring text to the user's browser from some other document and embed it in the tag's place. We have defined this as the <TXT SRC...> tag. It is not officially part of HTML, but we hope it will be.

OUR ENACTMENT

This site enacts the behavior of such a <TXT SRC...> tag. Any document placed on this server which contains a <TXT SRC...> tag will be presented in a Netscape or Microsoft browser as if the <TXT SRC...> tag were actually a part of HTML, bringing in the quoted material from any Web document, and embedding it in the tag's place.

IT ALSO WORKS EXTERNALLY

Not only that! If you want to use <TXT SRC...> in your own document, on some other server, linking to it THROUGH THIS SERVER will cause it to reach the user with the <TXT SRC...> tag correctly interpreted.

So come ahead, join our project! All you have to do is--

TRANSPUBLISH!

If you would like to try using the <TXT.SRC...> tag, arrange for us to link to it from here!

Send an e-mail with the URL of your document using <TXT SRC...> to: moretexts@crew.keio.sfc.ac.jp. Your document will be automatically listed here. (But don't forget-- don't transquote unless you have permission. You can easily transquote from things you've written yourself.))

SAMPLE USES OF TEXT TRANSPUBLISHING: xxxxx

On the World Wide Web, this already works technically for pictures (see our example***)
It is easy to transpublish a picture using the <IMG SRC...> tag of HTML. The <IMG SRC
tag brings in the picture from any other Web site and embeds it in the tag's place. All that
needed is the permission, and we're working to develop good "formats for transcopyright
permission".

THE NEED FOR A NEW TAG

Unfortunately, there is no similar way to transpublish an arbitrary portion of text. We thi
there is a need for a similar tag for text in HTML, like the picture tag, that will bring text t
the user's browser from some other document and embed it in the tag's place. We have
defined this as the <TXT SRC...> tag. It is not officially part of HTML, but we hope it wi
be.

OUR ENACTMENT

This site enacts the behavior of such a <TXT SRC...> tag. Any document placed on this
server which contains a <TXT SRC...> tag will be presented in a Netscape or Microsoft
browser as if the <TXT SRC...> tag were actually a part of HTML, bringing in the quoted
material from any Web document, and embedding it in the tag's place.

IT ALSO WORKS EXTERNALLY

Not only that! If you want to use <TXT SRC...> in your own document, on some other
server, linking to it THROUGH THIS SERVER will cause it to reach the user with the <TX
SRC...> tag correctly interpreted.

So come ahead, join our project! All you have to do is--

TRANSPUBLISH!

If you would like to try using the <TXT SRC...> tag, arrange for us to link to it from here!

Send an e-mail with the URL of your document using <TXT SRC...> to:
moretexts@crew.keio.sfc.ac.jp. Your document will be automatically listed here. (But don'
forget-- don't transquote unless you have permission. You can easily transquote from things
you've written yourself.))

SAMPLE USES OF TEXT TRANSPUBLISHING: xxxxx

ORIGINS

Ideas originally from Project Xanadu.

Transcopyright idea, design and formats by Ted Nelson.

<TXT SRC...> tag design by Andrew Pam of Xanadu Australia.

CGI implementation by Yousuke Igarashi.

Help and encouragement from: Marlene Mallicoat; Kazuhiro Yamada; Yoshihide Chubachi; CREW LAB of Keio SFC; Professor Hajime Ohiwa; and Dean Nobuo Saito.

HOzone
words words words words words words words words words words '''

para. templates ----------

edent 1-
flush at 1-
indent 1-
edent 1.4-
flush at 1.4-
indent at 1.4-
edent 1.6-
flush at 1.6-
indent at 1.6-
edent 2-
flush at 2-
indent at 2-

HO d2 ----------

EXPERIMENTAL SITE FOR TEXT TRANSPUBLISHING
virtual inclusion of material served from another document or site.
it may be possible for users easily to see

EXPERIMENTAL SITE FOR TEXT TRANSPUBLISHING

Text from Elsewhere

Bringing It In from Elsewhere: A TRANSPUBLICATION EXPERIMENT

(The rightsholder may even be paid for that part of the contribution, if certain parts of this project work out.)
the original context, which could be a benefit both to the reader and the original author or publisher. (If an appropriate system of micropayment can be developed, the original publisher may be paid as well for that quotation.)
Furthermore,

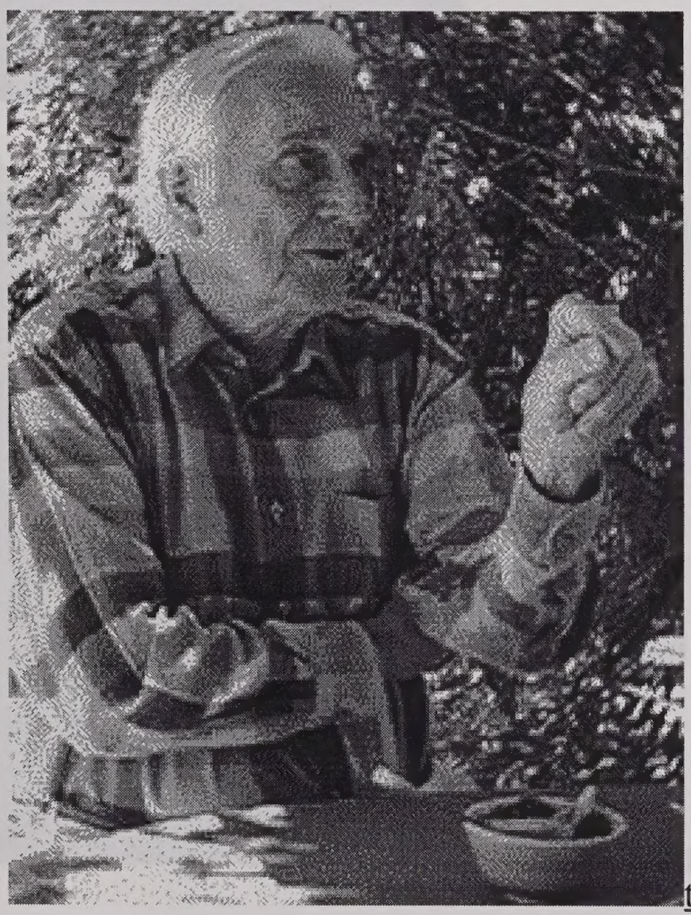
To Ted Nelson's Home Page

COMPUTER CELEBRITY PICTURES

To be available under transcopyright. (Real soon now.)

Transcopyright is a permission system, but methods for using it are still under development. For various transcopyright formats, see the permission page of the first picture. (Other permission pages for the following pictures are not up yet.)

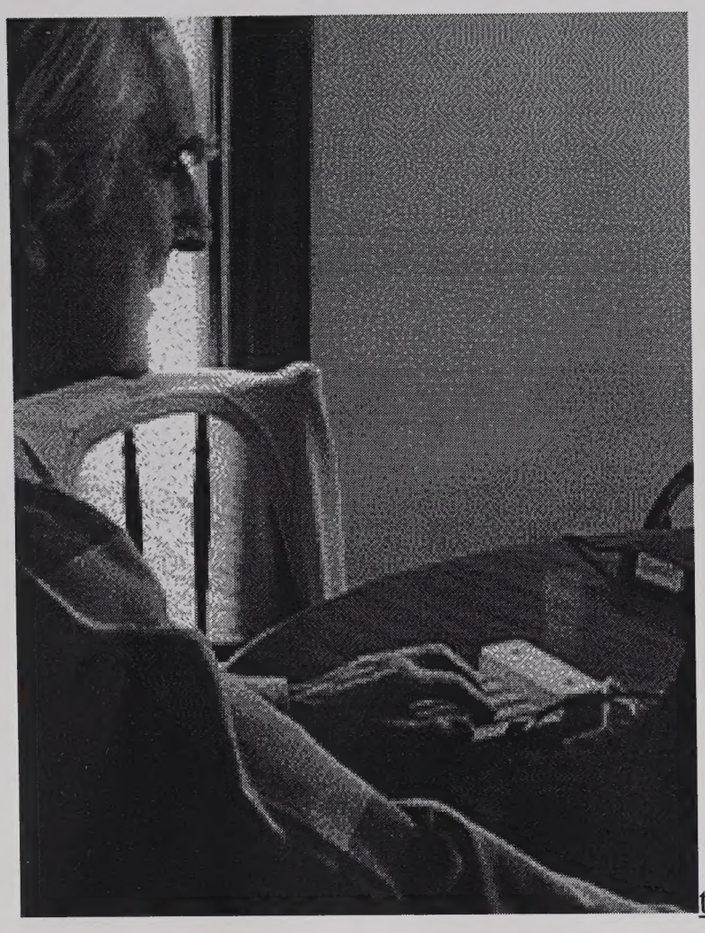
DOUGLAS C. ENGELBART, Bootstrap Foundation --



In his garden trans ©

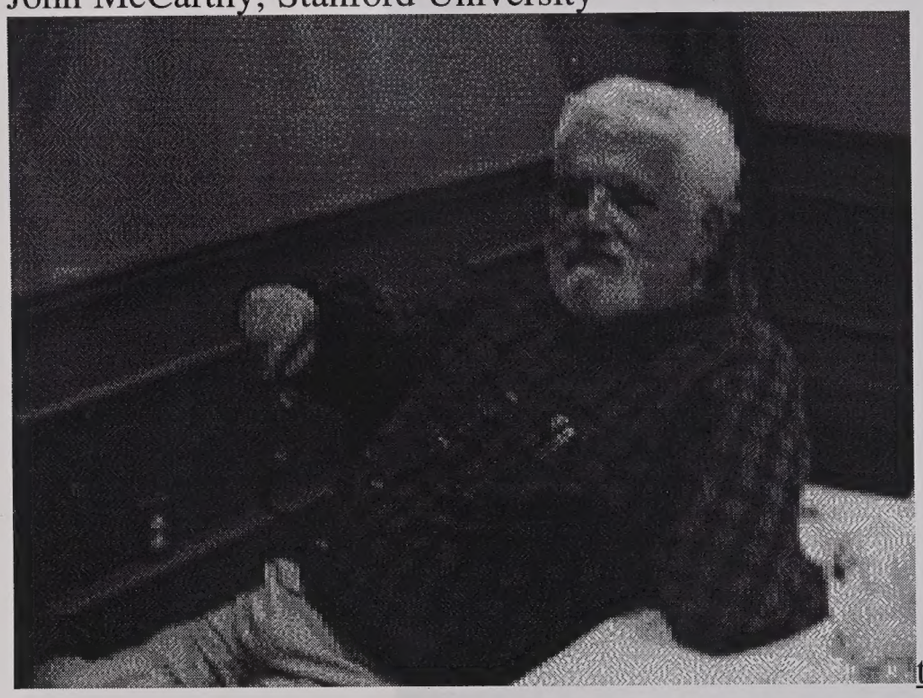
and at the famous chord keyboard

NEON DONE



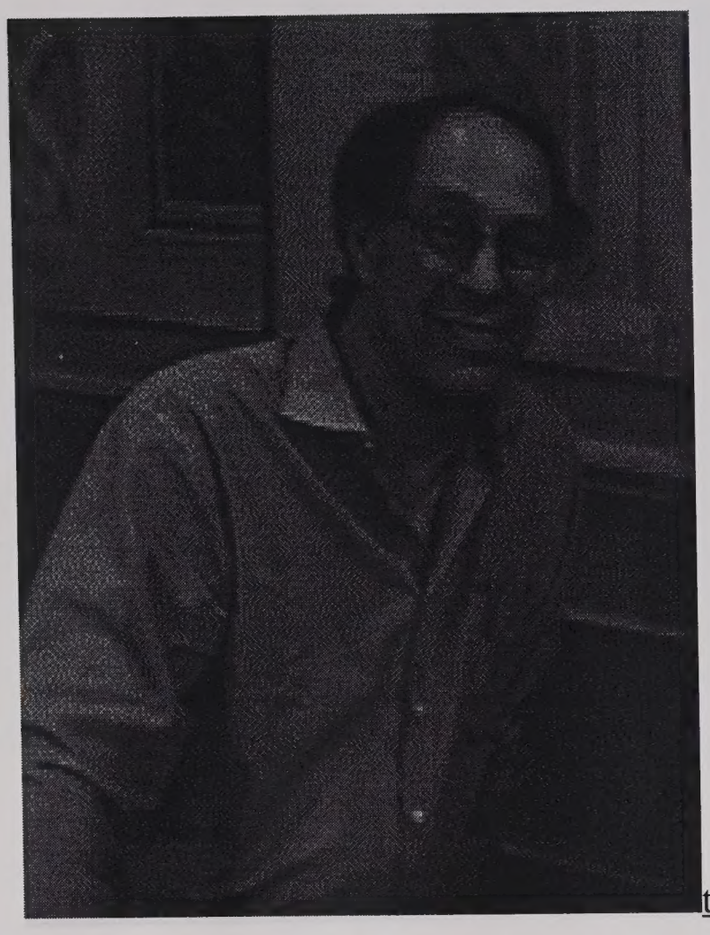
trans ©

John McCarthy, Stanford University

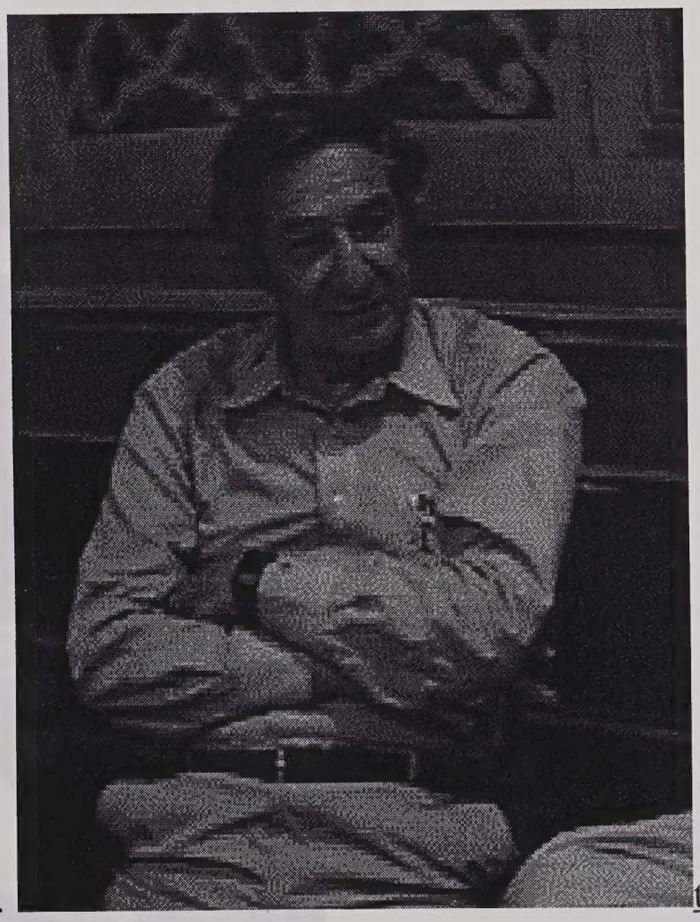


trans ©

DANNY BOBROW, Stanford University

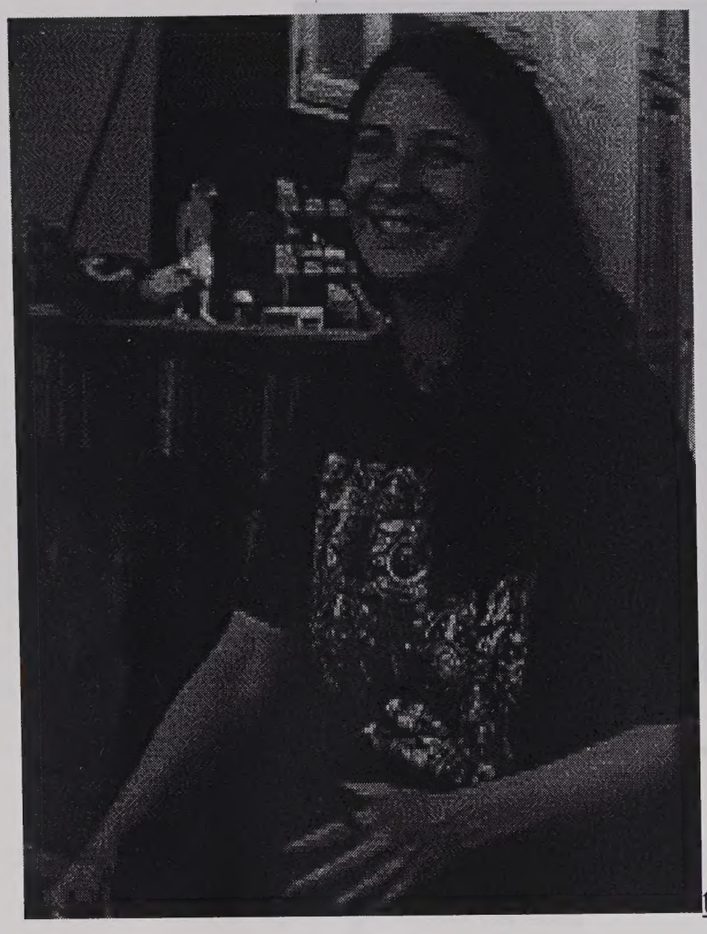


trans ©



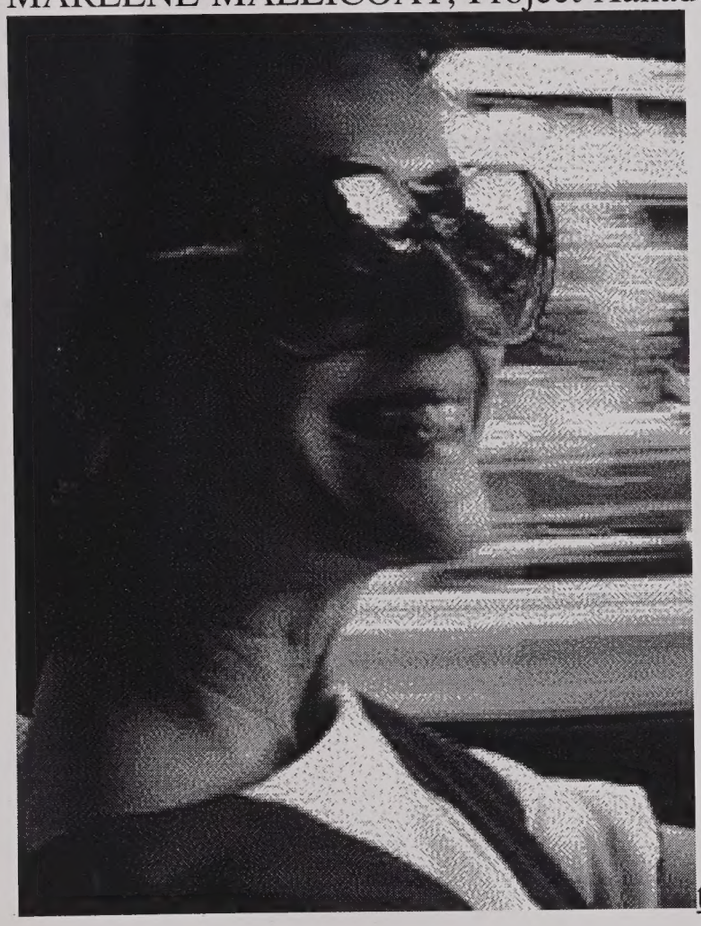
ED FREDKIN, M.I.T.

trans ©

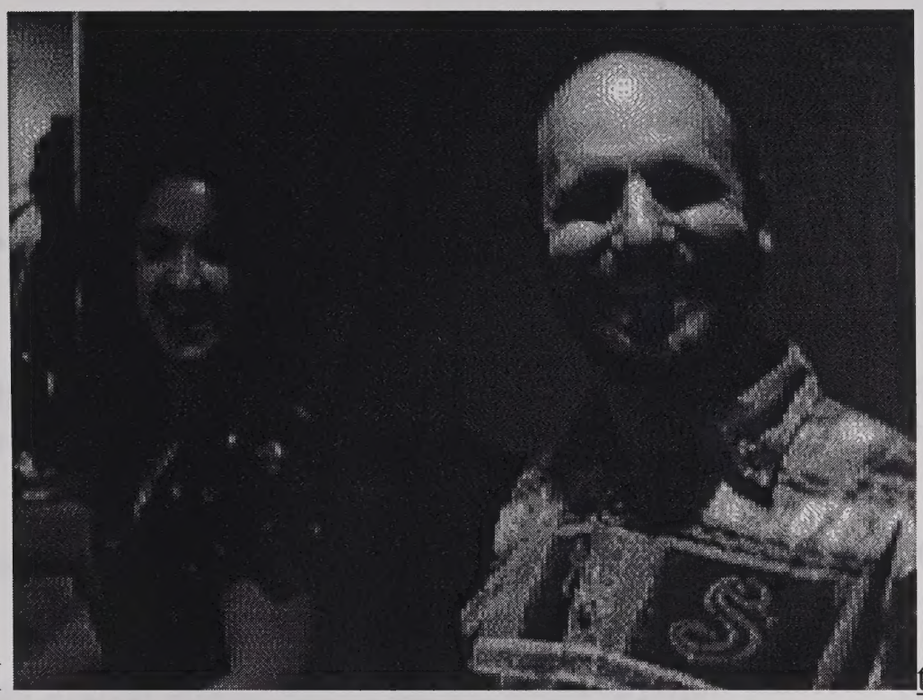


trans ©

MARLENE MALLICOAT, Project Xanadu.



trans ©

Ted Selker, IBM  trans ©

Still to come!

ROY STRINGER, Liverpool John Moores University

KATHERINE PHELPS, Glass Wings Press.

ANDREW PAM, Project Xanadu.

ROGER GREGORY, Project Xanadu.

TED NELSON, Project Xanadu. Photo by Kay Nishi.

KAY NISHI, ASCII Corporation.

ROHIT KHARE, W3 Consortium.

SAM EPSTEIN, Sensemedia.

transco.formats.html

96.12.07

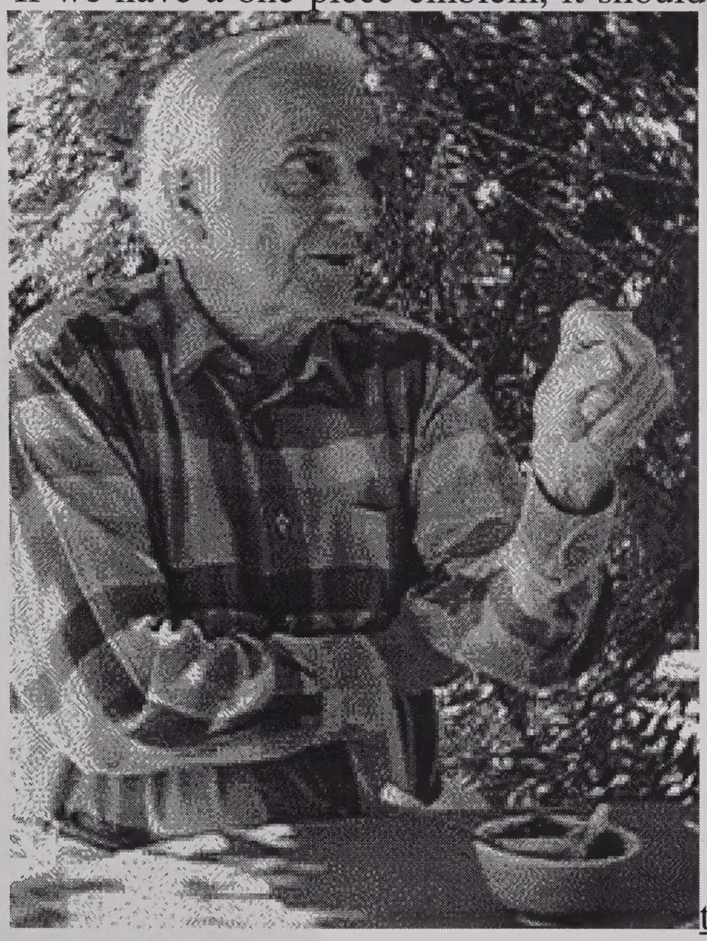
To Ted Nelson's Home Page

Different Transcopyright Formats

There are different possible transcopyright formats. The alternatives suggested here will give you some idea of the flexibility possible. I am suggesting these so you can see some alternatives.

(I would like to thank Doug Engelbart for his great patience in sitting still for all these alternatives.)

Let's look at four different logical arrangements of the transcopyright emblem. (Note that we can do better with symbols and artwork; right now we are just looking at the parts.)

One-piece emblem. If we have a one-piece emblem, it should probably link only to the permission page:  trans(c)

Two-piece emblem. A two-piece emblem allows us to link both to the permission

Different Transcopyright Formats

There are different possible transcopyright formats. The alternatives suggested here will give you some idea of the flexibility possible. I am suggesting these so you can see some alternatives.

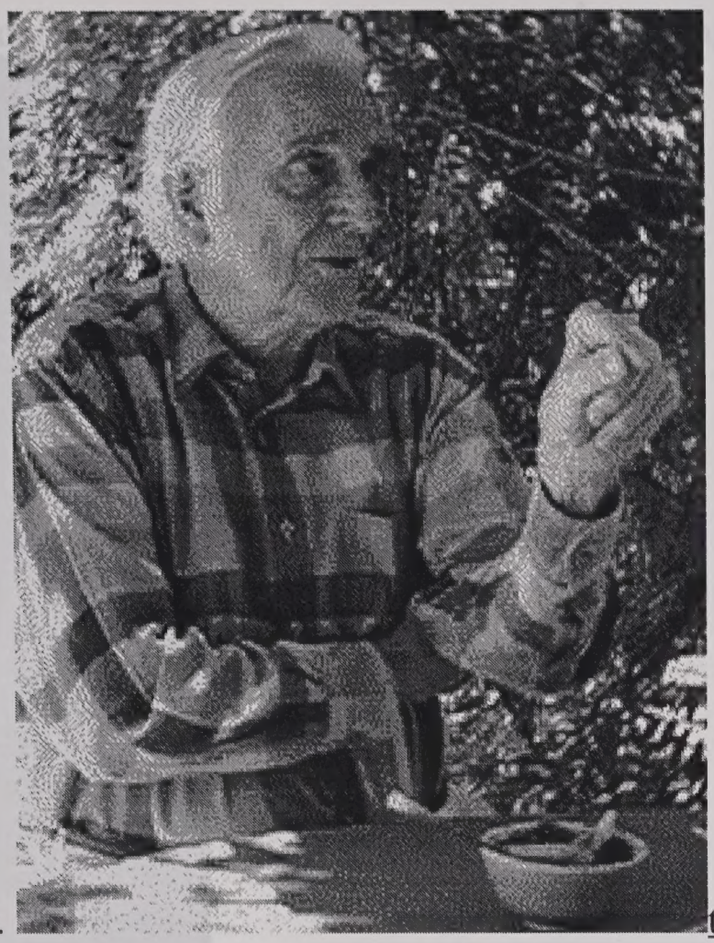
(I would like to thank Doug Engelbart for his great patience in sitting still for all these alternatives.)

Let's look at four different logical arrangements of the transcopyright emblem. (Note that we can do better with symbols and artwork; right now, we are just looking at the parts.)

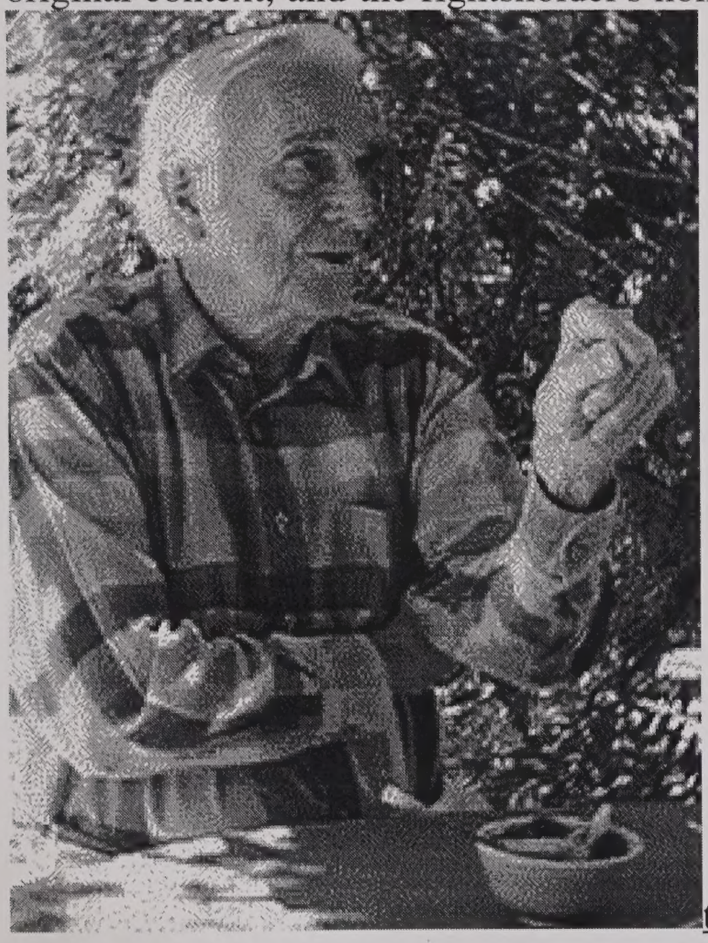
One-piece emblem. If we have a one-piece emblem, it should probably link only to

the permission page:

Two-piece emblem. A two-piece emblem allows us to link both to the permission

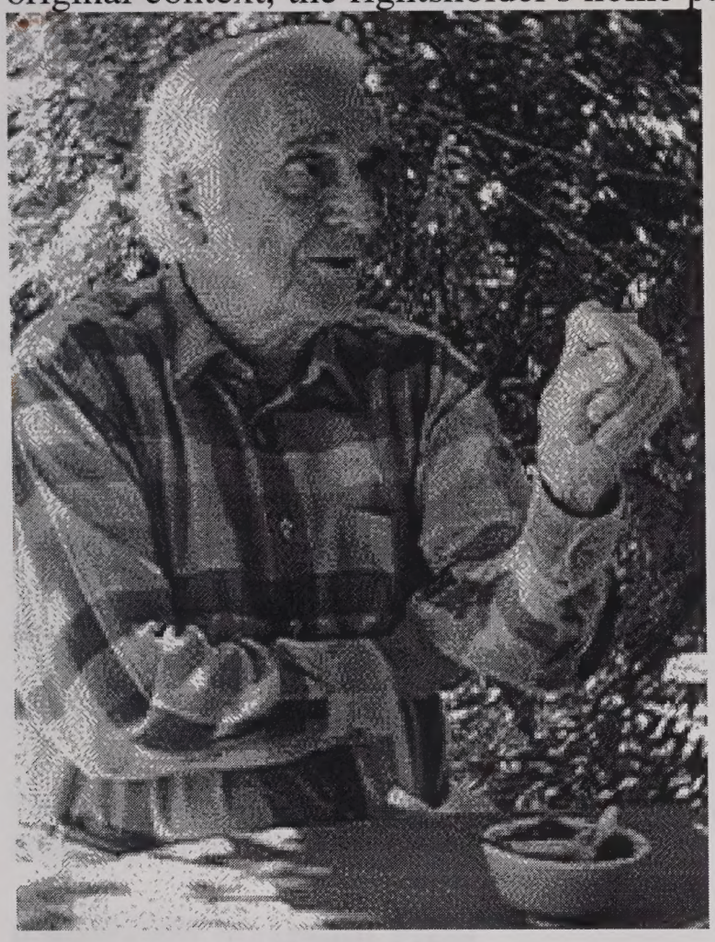
page and the original context.  trans (c)

Three-piece emblem. A three-piece emblem allows us to link to the permission page, the original context, and the rightsholder's home page.



trans © T.Nelson

original context, the rightsholder's home page, and the subject's home page.



Douglas Engelbart. trans © T.Nelson

STENOGRAPHIC ONLY

tco.pix.w.18 → w19

96.09.26 → 97.01.15 to Ted Nelson Home Page

Transcopyright for Pictures on the Web

(Prof.) Ted Nelson, Keio University Shonan Fujisawa Campus e-mail: ted@xanadu.net
home page http://sfc.keio.ac.jp.~/ted/

ÓTranscopyrightÓ is a new method of copyright for on-line publishers. It is actually a very compact way of giving permission for virtual, or indirect, re-use on line. It means: "Permission is given to republish this material virtually, under the condition that only an address pointer to this material is distributed by the republisher."

The name makes possible a compact notation. First consider ordinary copyright notation. A rightsholder (person or company) usually prints a copyright symbol next to their name, like:

©1996 Makeshift Press

On line, the publisher indicates transcopyright permission by making a small notational addition to the copyright symbol:

trans©1996 Makeshift Press

The additional ÓtransÓ has the effect of giving this permission for virtual re-use. (We may call this line the Òtranscopyright symbolÓ or Òshort form notice.Ó) Since it contains the original copyright notice, all the benefits of the original copyright notice are retained.

Note that no legislation is required for transcopyright to be legal; all that is required is for people to recognize this notation as permission to republish, and it will acquire status in law. (Much law, like this, is based on clarifying what people intend. Already there is legal standing for the term "shareware," which is also a permission doctrine.)

Right now transcopyright publishing can only be based on generosity. But if we can develop a payment system in extremely small denominations, each downloader can buy the items automatically from the rightsholder without caring about the very small cost.

INSTANT IMPLEMENTATION! For some data structures, such as text, transcopyright will require special implementation. But in the case of picture files in the commonest graphics format on the Web, no implementation is necessary. Use of transcopyright can begin immediately, because the mechanism already is built in.

The ordinary mechanism for Web publishing of a GIF or JPEG picture is simple. The picture is brought into a Web page simply by an html reference such as the following.

<IMG SRC="http://www.sfc.keio.ac.jp/~ted/Ted.pic"> This brings in the picture from

Transcopyright for Pictures on the Web

(Prof.) Ted Nelson, Keio University Shonan Fujisawa Campus e-mail: ted@xanadu.n
home page http://sfc.keio.ac.jp/~/ted/

"Transcopyright" is a new method of copyright for on-line publishers. It is actually a very compact way of giving permission for virtual, or indirect, re-use on line. It means "Permission is given to republish this material virtually, under the condition that only address pointer to this material is distributed by the republisher."

The name makes possible a compact notation. First consider ordinary copyright notation. A rightsholder (person or company) usually prints a copyright symbol next to their name, like:

©1996 Makesalit Press

On line, the publisher indicates transcopyright permission by making a small notational addition to the copyright symbol:

trans©1996 Makesalit Press

The additional "trans" has the effect of giving this permission for virtual re-use. (We may call this line the "transcopyright symbol" or "sixpence form notice.") Since it contains the original copyright notice, all the benefits of the original copyright notice are retained.

Note that no legislation is required for transcopyright to be legal; all that is required is for people to recognize this notation as permission to republish, and it will acquire status in law. (Much law, like this, is based on identifying what people intend. Already there is legal standing for the term "shareware," which is also a permission doctrine.)

Right now transcopyright publishing can only be based on generosity. But if we can develop a payment system in extremely small denominations, each downloader can buy the items automatically from the rightsholder without caring about the very small cost.

INSTANT IMPLEMENTATION! For some data structures, such as text, transcopyright will require special implementation. But in the case of picture files in the continuous graphics format on the Web, no implementation is necessary. Use of transcopyright can begin immediately, because the mechanism already is built in.

The ordinary mechanism for Web publishing of a GIF or JPEG picture is simple. The picture is brought into a Web page simply by an html reference such as the following.

<IMG SRC="http://www.sfc.keio.ac.jp/~ted/Ted.pic"> This brings in the picture from

that file. But in principle the image source can be on another computer anywhere else on the Internet. Even today, you could decorate your Web page with images pulled in real time from other peopleÕs servers all over the world.

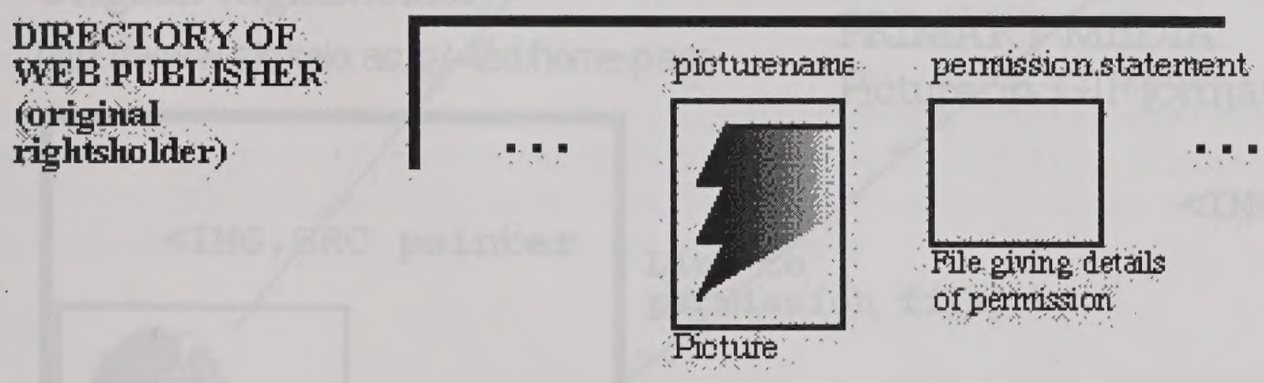
Why donÕt people do that? Partly because they donÕt have permission.

Permission. Permission is necessary because otherwise the rightsholder might object or bring lawsuit. Transcopyright provides that permission.

However, the definition of transcopyright given above has been made very general, since rightsholders will insist on choosing the details of their permission. The short form notice shown above is not sufficient to clarify the exact details of the permission being given in a particular case. Therefore these full details of the permission being granted by the rightsholder should be posted on line for each picture, in a place that is easy to find.

There are many possible details. For instance, the rightsholder may require that the republishing page have a pointer to the original page, labelled ÒTo original context.Ó Or the rightsholder might ask for a payment of some sort. However, since such a request is not likely to be honored on todayÕs Web, it should be stated as a polite request rather than a requirement.

Proposed permission file and its location. I propose that the simplest way to do this is to put the full specific permission statement into a permission file linked to the picture, and in the same directory.



Proposed complete format. What should the viewer see, and what should the links be?

I propose that the rightsholder should place the transcopyright symbol next to the picture on the original Web page, and make a link directly to the full permission statement from the transcopyright symbol.

I propose that the republisher should give acknowledgment to the rightsholder by placing the same visible copyright notice embedded in the new Web page, taken from the original Web page, and pointing to the same permission file. (Perhaps this should be a requirement.)

that file. But in principle the image source can be on another computer anywhere else on the Internet. Even today, you could decorate your Web page with images pulled in real time from other peoplesÕ servers all over the world.

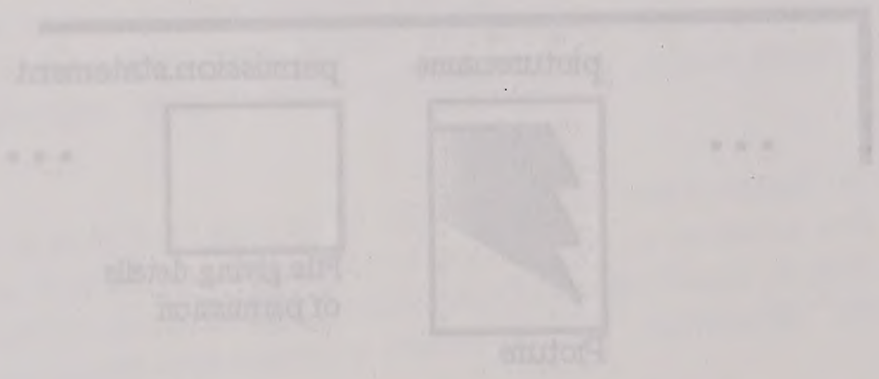
Why donÕt people do that? Partly because they donÕt have permission.

Permission. Permission is necessary because otherwise the rightsholder might object or bring lawsuit. Transcopyright provides that permission.

However, the definition of transcopyright given above has been made very general, since rightsholders will insist on choosing the details of their permission. The short form notice shown above is not sufficient to clarify the exact details of the permission being given in a particular case. Therefore these full details of the permission being granted by the rightsholder should be posted on line for each picture, in a place that is easy to find.

There are many possible details. For instance, the rightsholder may require that the republishing page have a pointer to the original page, labelled ÕTo original context.Õ Or the rightsholder might ask for a payment of some sort. However, since such a request is not likely to be honored on todayÕs Web, it should be stated as a polite request rather than a requirement.

Proposed permission file and its location. I propose that the simplest way to do this is to put the full specific permission statement into a permission file linked to the picture, and in the same directory.



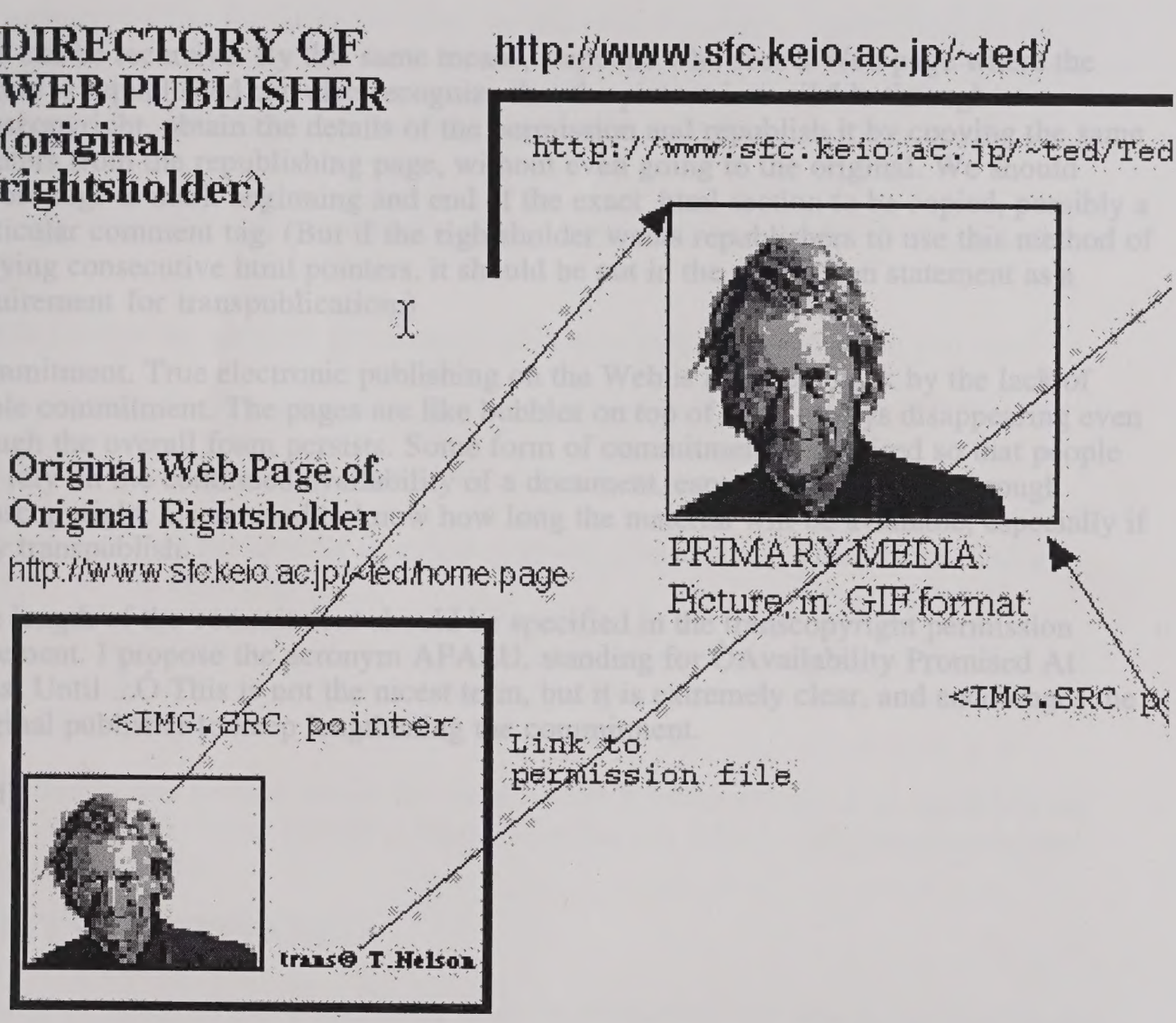
Proposed complete format. What should the viewer see, and what should the links be?

I propose that the rightsholder should place the transcopyright symbol next to the picture on the original Web page, and make a link directly to the full permission statement from the transcopyright symbol.

I propose that the republisher should give acknowledgment to the rightsholder by placing the same visible copyright notice embedded in the new Web page, taken from the original Web page, and pointing to the same permission file. (Perhaps this should be a requirement.)

The mechanism is shown below. (Next page.)

Luckily, this is easy. All the republisher needs to do is have the browser save the original page in .html format, and copy from the original page both the image source <IMG SRC...> and the publisherÕs transcopyright notice and link <A HREF...>, which points to the full permission. Since they are next to each other in the original .html file, only one copying operation is required. The following shows exactly what you would copy from the html source of my demonstration home page in order to use my transcopyrighted picture from my demo home page somewhere else on the Web-- but always delivered from my server.



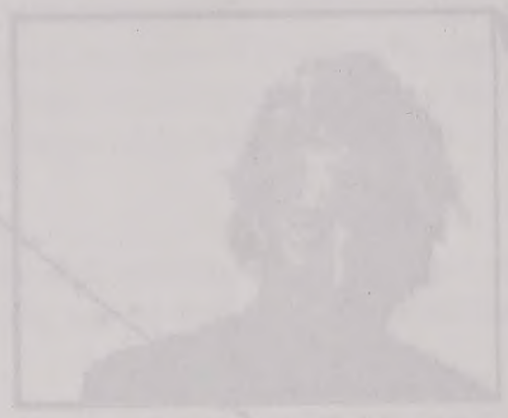
The mechanism is shown below. (Next page.)

Luckily, this is easy. All the republisher needs to do is have the browser save the original page in .html format, and copy from the original page both the image source <IMG SRC...> and the publisher's transcopyright notice and link <A HREF...>, which points to the full permission. Since they are next to each other in the original .html file, only one copying operation is required. The following shows exactly what you would copy from the html source of my demonstration home page in order to use my transcopyrighted picture from my demo home page somewhere else on the Web— but always delivered from my server.

DIRECTORY OF WEB PUBLISHER (original rightsholder)

http://www.sfc.keio.ac.jp/~ted/

http://www.sfc.keio.ac.jp/~ted/Ted



PRIMARY MEDIA
Picture in GIF format

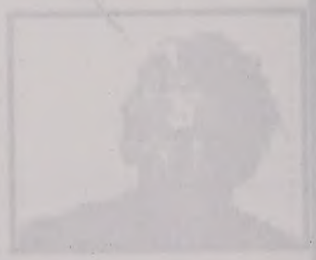
<IMG SRC pointer

Link to permission file

Original Web Page of Original Rightsholder

http://www.sfc.keio.ac.jp/~ted/home.page

<IMG SRC pointer



Ted T. Nelson

What the republisher copies into the new Web page (consecutive black stripe):

```
HREF="http://www.sfc.keio.ac.jp/~ted/alt.u.html">My Parallel Universe</A></H3>□□
<P><FONT SIZE=-1>This picture may be used elsewhere under Transcopyright□
permission.</FONT></P>□□<P><IMG SRC="Ted.nobg.gif" HEIGHT=216 WIDTH=256><FONT
SIZE=-1><A HREF="http://www.sfc.keio.ac.jp/~ted/Ted.pic.permish.html">Trans(c)1996□
T.Nelson</A></FONT></P>□□<P><FONT SIZE=-1>For the exact transcopyright permission,
follow the above□link.</FONT></P>□□</BODY>□</HTML>□
```

This can be recursive. By this same means, someone who sees a Web page where the picture is republished can also recognize that the picture is available through transcopyright, obtain the details of the permission and republish it by copying the same pointers from the republishing page, without even going to the original. We should choose tags to mark beginning and end of the exact .html section to be copied, possibly a particular comment tag. (But if the rightsholder wants republishers to use this method of copying consecutive html pointers, it should be put in the permission statement as a requirement for transpublication.)

Commitment. True electronic publishing on the Web is also held back by the lack of stable commitment. The pages are like bubbles on top of beer, always disappearing even though the overall foam persists. Some form of commitment is required so that people can rely on the continued availability of a document, especially for re-use through transcopyright. Users need to know how long the material will be available, especially if they transpublish.

The length of the commitment should be specified in the transcopyright permission statement. I propose the acronym APALU, standing for ÒAvailability Promised At Least Until ...Ó This is not the nicest term, but it is extremely clear, and encourages the original publisher to keep lengthening the commitment.

[end]

What the republisher copies into the new Web page (consecutive black stripe):

```
HREF="http://www.site.loc.gov/tas/alt.u.html">My Parallel Universe</A></H3>□□
<P><FONT SIZE=-1>This picture may be used elsewhere under Transcopyright
permission.</FONT>
```

```
</P>□□<P><FONT SIZE=-1>For the exact transcopyright permission,
follow the above□link.</FONT></P>□□</BODY>□</HTML>□□
```

This can be recursive. By this same means, someone who sees a Web page where the picture is republished can also recognize that the picture is available through transcopyright, obtain the details of the permission and republish it by copying the same pointers from the republishing page, without even going to the original. We should choose tags to mark beginning and end of the exact html section to be copied, possibly a particular comment tag. (But if the rightsholder wants republishers to use this method of copying consecutive html pointers, it should be put in the permission statement as a requirement for transpublication.)

Commitment. True electronic publishing on the Web is also held back by the lack of stable commitment. The pages are like bubbles on top of beer, always disappearing even though the overall foam persists. Some form of commitment is required so that people can rely on the continued availability of a document, especially for re-use through transcopyright. Users need to know how long the material will be available, especially if they transpublish.

The length of the commitment should be specified in the transcopyright permission statement. I propose the acronym APALU, standing for ÓAvailability Promised At Least Until ...Ó This is not the nicest term, but it is extremely clear, and encourages the original publisher to keep lengthening the commitment.

[end]

totpub.d6

96.10.04

To my Web site and transcopyright demo-- www./sfc.keio.ac.jp.~/ted/

(Prof.) Ted Nelson, Keio University Shonan Fujisawa Campus

e-mail: ted@xanadu.net

Toward Open Transpublishing

The principal objective of my work so far has been to create a viable medium for electronic network publishing with depth and permanence. Unfortunately the objectives have heretofore been difficult to explain. However, the success of the World Wide Web has shown the possibilities of network hypermedia publishing, and now people are listening better.

The objectives of Project Xanadu,* begun in 1960 (and so named in 1967), have always been:

1. A backbone for network hypertext publishing-- individual, anarchic and free;

2. general data structures for packages of digital media;

3. connection among these general data structures with very general links;

4. support as well for another principal relation, transclusion, which many find a different technical concept;

5. a change management system, representing the changes among these materials and links in a clean and useful manner, allowing previous states and connections to be replayed and reworked.

The rationale and general design for the different Xanadu systems is presented in my book *Literary Machines* (Mindful Press, Sausalito CA 94965, USA, version updated from time to time).

WHAT HAS HAPPENED

As most of us know, the World Wide Web (in its Technicolor version, assisted by the Mosaic browser) arrived suddenly about late 1994. It blindsided everybody. No one, not even its creators Tim Berners-Lee and Marc Andreessen, was prepared for the impact that has occurred. For us at the Xanadu project, however, the surprise was not the spectacular growth of network hypertext publishing, but the simple-mindedness of the system that accomplished it. And so we were as astonished as everyone else, though we had predicted exactly the scale of growth that the Web has enjoyed.

totpub.d6

96.10.04

To my Web site and transcopyright demo– www.sfc.keio.ac.jp/~ted/

(Prof.) Ted Nelson, Keio University Shonan Fujisawa Campus

e-mail: ted@xanadu.net

Toward Open Transpublishing

The principal objective of my work so far has been to create a viable medium for electronic network publishing with depth and permanence. Unfortunately the objectives have heretofore been difficult to explain. However, the success of the World Wide Web has shown the possibilities of network hypermedia publishing, and now people are listening better.

The objectives of Project Xanadu,* begun in 1960 (and so named in 1967), have always been:

1. A backbone for network hypertext publishing– individual, anarchic and free;

2. general data structures for packages of digital media;

3. connection among these general data structures with very general links;

4. support as well for another principal relation, transclusion, which many find a different technical concept;

5. a change management system, representing the changes among these materials and links in a clean and useful manner, allowing previous states and connections to be replayed and reworked.

The rationale and general design for the different Xanadu systems is presented in my book *Literary Machines* (Mindful Press, Sausalito CA 94965, USA, version updated from time to time).

WHAT HAS HAPPENED

As most of us know, the World Wide Web (in its Technicolor version, assisted by the Mosaic browser) arrived suddenly about late 1994. It blindsided everybody. No one, not even its creators Tim Berners-Lee and Marc Andreessen, was prepared for the impact that has occurred. For us at the Xanadu project, however, the surprise was not the spectacular growth of network hypertext publishing, but the simple-mindedness of the system that accomplished it. And so we were as astonished as everyone else, though we had predicted exactly the scale of growth that the Web has enjoyed.

THE TRUE STRUCTURE OF INFORMATION

People say, "The World Wide Web is so disorganized!" This shows that they do not understand the structure of information.

Much of the problem we have experienced is based on a paradigm confrontation between the Xanadu project and the rest of the computer field. The rest of the computer field believes that information should be hierarchical and sharply divided among its parts. This arises from what I consider to be a strange and misguided idea of information.

Information is not intrinsically neat. It is intrinsically messy. The information of every field is really what I would like to call a ***perplex*** ** of relationships, disagreements, alternatives and uncertainties. That is what the information is really like.

However, we have been misled by libraries, college courses, and some sort of tradition, to think that information is really hierarchical, clear and exact. Therefore systems for information storage and presentation-- like database and word processing-- are designed to hold only what is hierarchical, clear and exact, and not the real information of the world.

I maintain that this has based our computer world on a complete misconception. Hierarchical, clear and exact data is a rare special case, trivial in comparison with the real complexity of information, and yet we have programmed our systems only for this trivial special case. I maintain that the real problem is to represent the perplexes of thought in ways that are as close as possible to the exact representation of their complex structure, with all the variants and alternatives present.

Interestingly, we alrady have such a system. It is called literature. The structure of literature, including all academic literatures, represents the different viewpoints of many participants and commentators.

Thus I see the real issue in computer software as representing relationships and alternatives correctly, in a new kind of electronic literature.

This the Xanadu project has endeavored to do, allowing the same material to exist simultaneously in many different contexts and to be reworked in many different ways, with links and comments made by all parties.

SHORTCOMINGS OF THE WEB

After the huzzahs of last year, we note creeping difficulties with the World Wide Web.***

Links can be followed only in one direction; parties who link to a document from outside are disenfranchised, since no one can see their links from inside that document. And the published materials disappear unpredictably; there is no knowing what will last. The World Wide Web is like a foam of ever-popping bubbles.

THE TRUE STRUCTURE OF INFORMATION

People say, "The World Wide Web is so disorganized!" This shows that they do not understand the structure of information.

Much of the problem we have experienced is based on a paradigm confrontation between the Xanadu project and the rest of the computer field. The rest of the computer field believes that information should be hierarchical and sharply divided among its parts. This arises from what I consider to be a strange and misguided idea of information.

Information is not intrinsically neat. It is intrinsically messy. The information of every field is really what I would like to call a *perplex* ** of relationships, disagreements, alternatives and uncertainties. That is what the information is really like.

However, we have been misled by libraries, college courses, and some sort of tradition, to think that information is really hierarchical, clear and exact. Therefore systems for information storage and presentation-- like database and word processing-- are designed to hold only what is hierarchical, clear and exact, and not the real information of the world.

I maintain that this has based our computer world on a complete misconception. Hierarchical, clear and exact data is a rare special case, trivial in comparison with the real complexity of information, and yet we have programmed our systems only for this trivial special case. I maintain that the real problem is to represent the perplexes of thought in ways that are as close as possible to the exact representation of their complex structure, with all the variants and alternatives present.

Interestingly, we already have such a system. It is called literature. The structure of literature, including all academic literatures, represents the different viewpoints of many participants and commentators.

Thus I see the real issue in computer software as representing relationships and alternatives correctly, in a new kind of electronic literature.

This the Xanadu project has endeavored to do, allowing the same material to exist simultaneously in many different contexts and to be reworked in many different ways, with links and comments made by all parties.

SHORTCOMINGS OF THE WEB

After the huzzahs of last year, we note creeping difficulties with the World Wide Web.***

Links can be followed only in one direction; parties who link to a document from outside are disenfranchised, since no one can see their links from inside that document. And the published materials disappear unpredictably; there is no knowing what will last. The World Wide Web is like a foam of ever-popping bubbles.

We are presently caught in a stage that the Xanadu Project foresaw and hoped to bypass.

THE OLD XANADU

The old Xanadu (1960-1992) consisted of a series of designs, several of which were partially implemented, and the last of which is now in beta testing.****

The old Xanadu was intended to be a service run like Compuserve-like business service, which all Xanadu users would connnect to. The users would be able to publish not only their own documents, but annotations to anyone else's documents, and annotations to each others' annotations, and so on. All these links could be followed in either direction.

TRANSCLUSION AND TRANSPUBLICATION

But of equal importance to links, we believed, is the mechanism we call transclusion, the equivalent of one thing in many places. We believe that this is vital for re-use, understanding, and the solution of the copyright problem.

Transclusion means virtual inclusion with reference to an original.*****

The mechanism of transclusion has two special benefits.

1) Transclusion can be used to define a fair royalty payment, whereby a user pays automatically for all material downloaded, in small amounts, pro rata. (The user then is entitled to keep these materials, just as one keeps a book.)

2) Transclusion also allows anyone to quote, re-use and revise any document in the system, since this is to be done by transclusion; the result is a virtual document, a list of the pieces to be downloaded (along with new material), and so each portion is bought from its original publisher (usually the author). Thus the apparent conflict between copyright and user freedom was to be solved.

Transpublishing means participating in a copyright system that allows this unrestricted quotation, re-use and revision of copyrighted materials by transclusion. We also believe that it may solve much of the copyright problem.

THE NEW XANADU

The old Xanadu project essentially ended in 1992, when Autodesk ended support of the server we were developing.

Since then I have been working on the new Xanadu project, with exactly the same objectives, and also using the lynchpin of transclusion for the same objective: ***transpublishing*** , where automatic payment to the publisher will allow unrestricted quotation, re-use and revision of any portions of material.

We are presently caught in a stage that the Xanadu Project foresaw and hoped to bypass.

THE OLD XANADU

The old Xanadu (1960-1992) consisted of a series of designs, several of which were partially implemented, and the last of which is now in beta testing.****

The old Xanadu was intended to be a service run like Compuserve-like business service, which all Xanadu users would connect to. The users would be able to publish not only their own documents, but annotations to anyone else's documents, and annotations to each others' annotations, and so on. All these links could be followed in either direction.

TRANSCLUSION AND TRANSPUBLICATION

But of equal importance to links, we believed, is the mechanism we call transclusion, the equivalent of one thing in many places. We believe that this is vital for re-use, understanding, and the solution of the copyright problem.

Transclusion means virtual inclusion with reference to an original.*****

The mechanism of transclusion has two special benefits.

1) Transclusion can be used to define a fair royalty payment, whereby a user pays automatically for all material downloaded, in small amounts, pro rata. (The user then is entitled to keep these materials, just as one keeps a book.)

2) Transclusion also allows anyone to quote, re-use and revise any document in the system, since this is to be done by transclusion; the result is a virtual document, a list of the pieces to be downloaded (along with new material), and so each portion is bought from its original publisher (usually the author). Thus the apparent conflict between copyright and user freedom was to be solved.

Transpublishing means participating in a copyright system that allows this unrestricted quotation, re-use and revision of copyrighted materials by transclusion. We also believe that it may solve much of the copyright problem.

THE NEW XANADU

The old Xanadu project essentially ended in 1992, when Autodesk ended support of the server we were developing.

Since then I have been working on the new Xanadu project, with exactly the same objectives, and also using the lynchpin of transclusion for the same objective: transpublishing, where automatic payment to the publisher will allow unrestricted quotation, re-use and revision of any portions of material.

However, in today's new Internet environment, the old Xanadu has been turned inside out. Instead of a closed service, it is a system that will use the Internet and work in a simplified fashion.

The new Xanadu consists of several public-domain ideas:

1. ***Transcopyright***, an easy-to-use permission system for allowing transclusive re-use.******

2. **OSMIC** (Open Standard for Media InterConnection), a set of specifications for the exchange of complex linked documents.*******

OSMIC is intended to represent interconnection better than any system now available.

At the present time there is no standard among hypertext and hypermedia systems. Such products as HyperCard, Director, FrameMaker, HyperWave, Multicosm and Hyper Link Modeler all offer some kind of connections amongst the data. However, all of them offer different kinds of interconnection amongst the data, so that they cannot be used compatibly, and their data cannot be exchanged.

The World Wide Web has become a kind of a standard for linkage, but its linkage leaves much to be desired, as felt by more and more people now.

I have specified the OSMIC standard as a possible solution. This is a set of specifications for describing a complex linked document for interchange and publication.

The OSMIC links are much more general than those of HTML or any other product. They are followable in both directions and allow arbitrary endsets They may also be assigned any number of types. I believe productions and data which use any of the existing hypermedia software may be translated into OSMIC form with no loss.

Thus OSMIC has a chance of becoming a behind-the-scenes standard for connected structure. The Web can then become a projection screen for the more general, stable structures possible using OSMIC.

Thus OSMIC is intended as a way to express generalized complex relations-- indeed, perplexes.

It also has a notation which clears up issues of priority, origin-- and leaves links in place.

To clarify its origins, OSMIC is essentially a public-domain Xanadu. It takes the generalized relationships that Xanadu made possible and expresses them in a technology-free fashion.

3. Many-to-many micropayment.

This will allow any members of the public to offer their writings, pictures, music and

However, in today's new Internet environment, the old Xanadu has been turned inside out. Instead of a closed service, it is a system that will use the Internet and work in a simplified fashion.

The new Xanadu consists of several public-domain ideas:

1. **Transcopyright**, an easy-to-use permission system for allowing transclusive re-use.*****

2. OSMIC (Open Standard for Media InterConnection), a set of specifications for the exchange of complex linked documents.******

OSMIC is intended to represent interconnection better than any system now available.

At the present time there is no standard among hypertext and hypermedia systems. Such products as HyperCard, Director, FrameMaker, HyperWave, Multicosm and Hyper Link Modeler all offer some kind of connections amongst the data. However, all of them offer different kinds of interconnection amongst the data, so that they cannot be used compatibly, and their data cannot be exchanged.

The World Wide Web has become a kind of a standard for linkage, but its linkage leaves much to be desired, as felt by more and more people now.

I have specified the OSMIC standard as a possible solution. This is a set of specifications for describing a complex linked document for interchange and publication.

The OSMIC links are much more general than those of HTML or any other product. They are followable in both directions and allow arbitrary endsets They may also be assigned any number of types. I believe productions and data which use any of the existing hypermedia software may be translated into OSMIC form with no loss.

Thus OSMIC has a chance of becoming a behind-the-scenes standard for connected structure. The Web can then become a projection screen for the more general, stable structures possible using OSMIC.

Thus OSMIC is intended as a way to express generalized complex relations-- indeed, perplexes.

It also has a notation which clears up issues of priority, origin-- and leaves links in place.

To clarify its origins, OSMIC is essentially a public-domain Xanadu. It takes the generalized relationships that Xanadu made possible and expresses them in a technology-free fashion.

3. Many-to-many micropayment.

This will allow any members of the public to offer their writings, pictures, music and

videos for sale under transcopyright, and allow any members of the public to buy them. Ideally such a system should use standard Web browsers and be applicative, usable on top of existing services as well as for the sale of media data and OSMIC materials.

PROPRIETARY ASPECTS OF THE NEW XANADU

The new Xanadu is also, like the old, intended to be a network of licensed or franchised document servers.

However, the services have been redefined to fit today's Web and Internet. Specifically, they will provide:

1) Publish-by-Name service;

2) Long-Term Committed publishing;

3) Distribution of document storage for efficiency.

CONCLUSION

I still believe that transpublishing offers a unique opportunity for human understanding, learning and culture. Far from being defeated by the World Wide Web, I believe it is within reach.

---

* "Xanadu" is a service mark and registered trade mark of Project Xanadu, 3020 Bridgeway #295, Sausalito CA 94965, USA.

** Frederick C. Crews, *The Pooh Perplex* .

*** Andrew Pam, "Where the World Wide Web Went Wrong." Available on the Internet.

**** Keith Henson, XOC, Inc., personal communication.

***** Or proxy or cache which reports to the original and is therefore not distinguishable from it.

****** See explanations and a demonstration of transcopyright at www.sfc.keio.ac.jp/~ted/.

******* Specifications for OSMIC will be posted to my Web site.

videos for sale under transcopyright, and allow any members of the public to buy them. Ideally such a system should use standard Web browsers and be applicative, usable on top of existing services as well as for the sale of media data and OSMIC materials.

PROPRIETARY ASPECTS OF THE NEW XANADU

The new Xanadu is also, like the old, intended to be a network of licensed or franchised document servers.

However, the services have been redefined to fit today's Web and Internet. Specifically, they will provide:

1) Publish-by-Name service;

2) Long-Term Committed publishing;

3) Distribution of document storage for efficiency.

CONCLUSION

I still believe that transpublishing offers a unique opportunity for human understanding, learning and culture. Far from being defeated by the World Wide Web, I believe it is within reach.

---

* "Xanadu" is a service mark and registered trade mark of Project Xanadu, 3020 Bridgeway #295, Sausalito CA 94965, USA.

** Frederick C. Crews, The Pooh Perplex.

*** Andrew Pam, "Where the World Wide Web Went Wrong." Available on the Internet.

**** Keith Henson, XOC, Inc., personal communication.

***** Or proxy or cache which reports to the original and is therefore not distinguishable from it.

****** See explanations and a demonstration of transcopyright at www.sfc.keio.ac.jp/~ted/.

******* Specifications for OSMIC will be posted to my Web site.

KEIO TRANSPUBLISHING PROJECT • Profs. Ohiwa, Naemura, Nelson • Students: Mr. Igarashi and others • Independent Programmer: Mr. Togasi

TRANSPUBLISHING

CAN IT HELP SOLVE THE COPYRIGHT PROBLEM?

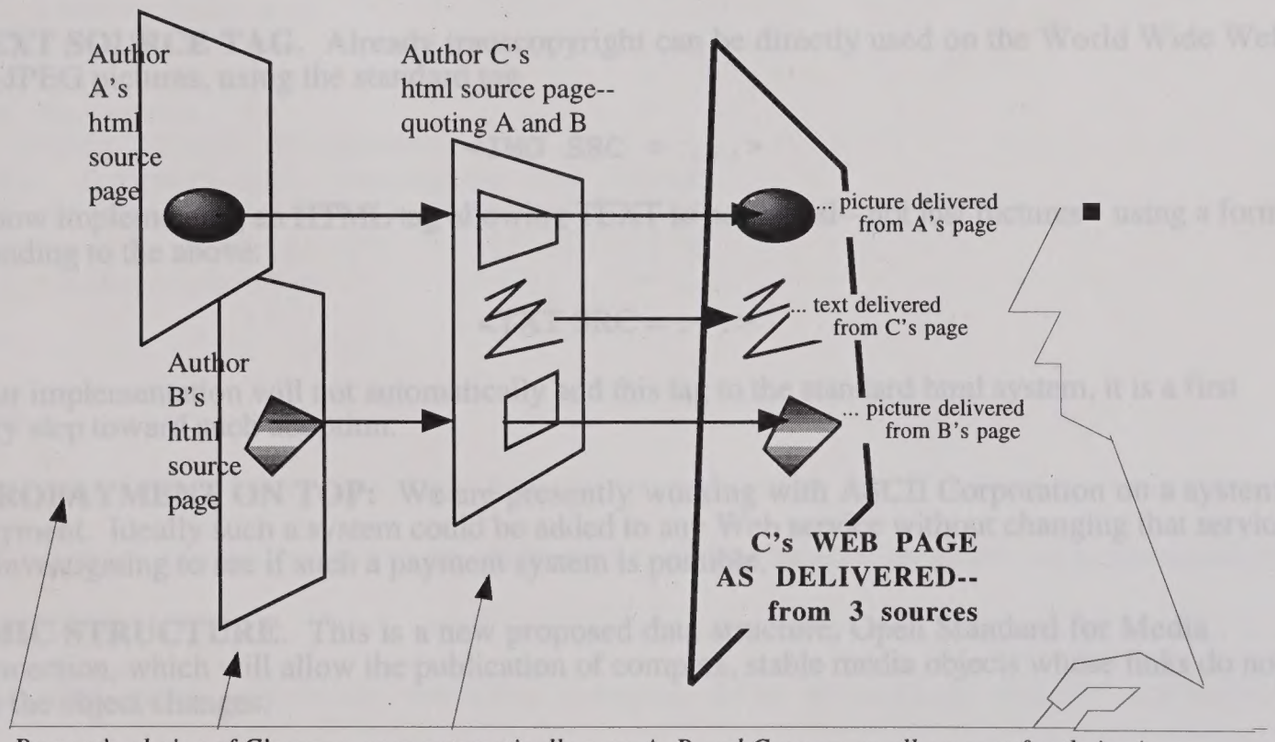
The copyright system creates a conflict between two desirable ideas.

(1) Authors, artists and publishers want praise and profit for their writings, pictures and other creative work.

But (2) the public wants to be able to re-use copyrighted materials in ways that they find beneficial and appropriate.

TRANSPUBLISHING is a possible Win-Win solution that combines both desirable features: contents may be freely re-used and republished in new arrangements by anyone, while the authors and publishers receive credit (and possibly money). It only works on computer networks (like the Internet).

In transpublishing, an electronic document quotes from another electronic document by pointing at the part to be quoted. Each quotation is delivered to the user from its original (or some equivalent, owned by its originator). This means that the user's payment for that part (if there is any payment) goes to the originator.



By user's choice of C's page, user automatically pays A, B and C a very small amount for their pieces.

TRANSCOPYRIGHT is a new legal arrangement, a form of permission allowing people to re-use material in this way on the network. "Trans©" is a compact way of saying, "I permit you to re-use this material in some new context, if it is always delivered from where I say." This allows the material to be re-used in new contexts with credit and payment to the originator.

KEIO TRANSPUBLISHING PROJECT • Prof. Ohira, Nagamura, Nielson • Students: Mr. Igarashi and others
• Independent Programmer: Mr. Togasi

TRANSPUBLISHING

CAN IT HELP SOLVE THE COPYRIGHT PROBLEM?

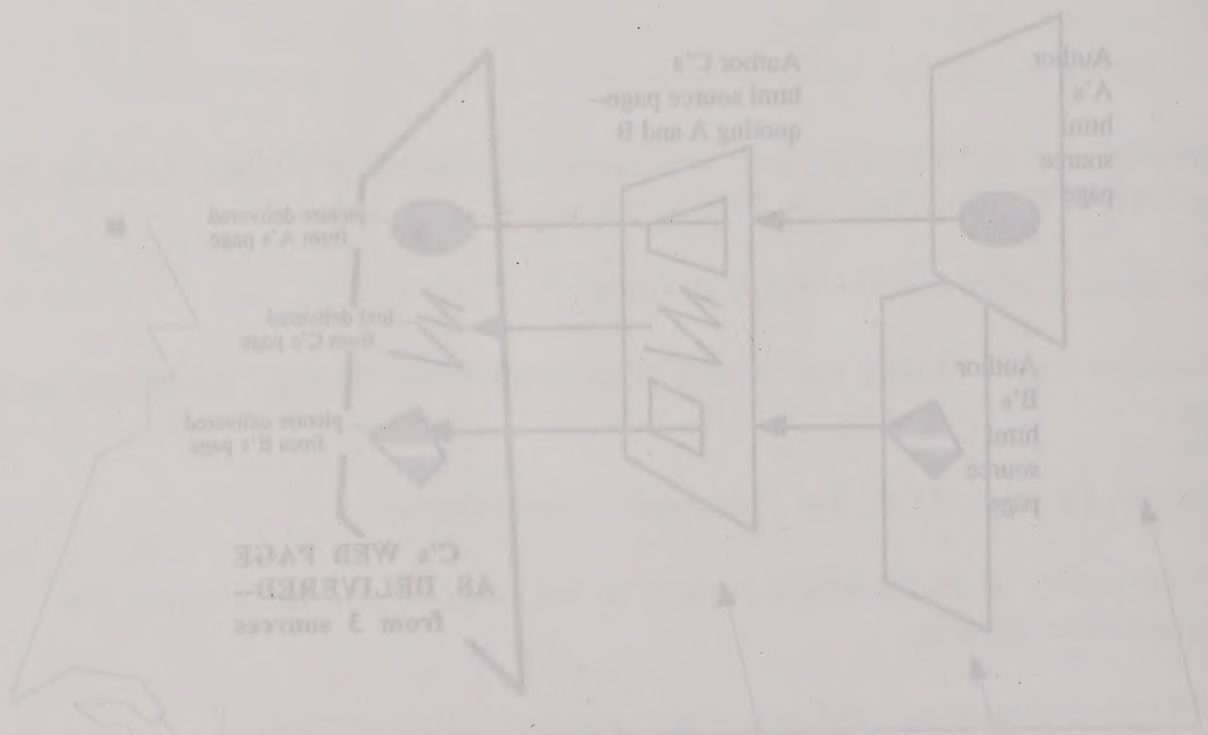
The copyright system creates a conflict between two desirable ideas.

(1) Authors, artists and publishers want praise and profit for their writings, pictures and other creative work.

But (2) the public wants to be able to re-use copyrighted materials in ways that they find beneficial and appropriate.

TRANSPUBLISHING is a possible Win-Win solution that combines both desirable features: contents may be freely re-used and republished in new arrangements by anyone, while the authors and publishers receive credit (and possibly money). It only works on computer networks (like the Internet).

In transpublishing, an electronic document quotes from another electronic document by pointing at the part to be quoted. Each quotation is delivered to the user from its original (or some equivalent, owned by its originator). This means that the user's payment for that part (if there is any payment) goes to the originator.



By user's click on C's page, user automatically pays A, B and C a very small amount for their pieces.

TRANSCOPYRIGHT is a new legal arrangement, a form of permission allowing people to re-use material in this way on the network. "Trans©" is a compact way of saying, "I permit you to re-use this material in some new context, if it is always delivered from where I say." This allows the material to be re-used in new contexts with credit and payment to the originator.

KEIO TRANSPUBLISHING PROJECT

We are working on a number of aspects of electronic publishing necessary for tomorrow's on-line document systems.

1. TRANSCOPYRIGHT: a form of permission allowing people to re-use and republish material virtually on the network."Trans©" is a compact way of saying, "I permit you to re-use this material in some new context, if it is delivered from my server." This allows credit and payment to the originator.

See our Web pages at

www.sfc.keio.ac.jp/~ted/

2. But transcopyright leaves a lot of options for the rightsholder. The proposed "**KEIO TRANSCOPYRIGHT FORMATS**" will provide suggestions for various specific details of transcopyright permission.

This will include a simplified permission statement and a transcopyright notice divided into two pieces, one pointing to the permission statement and one pointing to the original context. It will also provide suggested transcopyright formats for other data types, especially sound and video.

3. A TEXT SOURCE TAG. Already transcopyright can be directly used on the World Wide Web for GIF and JPEG pictures, using the standard tag

<IMG SRC = ...>

We are now implementing an HTML tag allowing TEXT to be quoted-- not just pictures-- using a format corresponding to the above:

<TXT SRC = . . .>

While our implementation will not automatically add this tag to the standard html system, it is a first necessary step toward such adoption.

4. MICROPAYMENT ON TOP: We are presently working with ASCII Corporation on a system for micropayment. Ideally such a system could be added to any Web service without changing that service. We are investigating to see if such a payment system is possible.

5. OSMIC STRUCTURE. This is a new proposed data structure, Open Standard for Media InterConnection, which will allow the publication of complex, stable media objects whose links do not break as the object changes.

6. PUBLISH-BY-NAME (URN). Today, publication on the Web makes a document synonymous with its address (URL). This is unfortunate. A "publish-by-name" system will give the object a unique nameplex (title + author + time + node of origin), sometimes called a Universal Resource Name or URN. This will allow the object to be put at many different locations, both for efficiency of delivery and safety of the contents.

7. STABLE PUBLISHING WITH LONG-TERM COMMITMENT. Today's Internet publishing is temporary and unreliable; there is no way to make a believable promise that a document will be available for 50 years or longer. We are investigating the organizational arrangements that will make such promises possible and credible.

KEIO TRANSPUBLISHING PROJECT

We are working on a number of aspects of electronic publishing necessary for tomorrow's on-line document systems.

1. TRANSCOPYRIGHT: a form of permission allowing people to re-use and republish material virtually on the network. "TransC" is a compact way of saying, "I permit you to re-use this material in some new context, if it is delivered from my server." This allows credit and payment to the originator.

See our Web pages at

www.sfc.keio.ac.jp/~ted/

2. But transcopyright leaves a lot of options for the rightsholder. The proposed **"KEIO TRANSCOPYRIGHT FORMATS"** will provide suggestions for various specific details of transcopyright permission.

This will include a simplified permission statement and a transcopyright notice divided into two parts, one pointing to the permission statement and one pointing to the original context. It will also provide suggested transcopyright formats for other data types, especially sound and video.

3. A TEXT SOURCE TAG. Already transcopyright can be directly used on the World Wide Web for GIF and JPEG pictures, using the standard tag

<IMG SRC = ...>

We are now implementing an HTML tag allowing TEXT to be quoted— not just pictures— using a format corresponding to the above:

<TXT SRC = ...>

While our implementation will not automatically add this tag to the standard html system, it is a first necessary step toward such adoption.

4. MICROPAYMENT ON TOP: We are presently working with ASCII Corporation on a system for micropayment. Ideally such a system could be added to any Web service without changing that service. We are investigating to see if such a payment system is possible.

5. OSMIC STRUCTURE. This is a new proposed data structure, Open Standard for Media InterConnection, which will allow the publication of complex, stable media objects whose links do not break as the object changes.

6. PUBLISH-BY-NAME (URN). Today, publication on the Web makes a document synonymous with its address (URL). This is unfortunate. A "publish-by-name" system will give the object a unique namepiex (title + author + time + node of origin), sometimes called a Universal Resource Name or URN. This will allow the object to be put at many different locations, both for efficiency of delivery and safety of the contents.

7. STABLE PUBLISHING WITH LONG-TERM COMMITMENT. Today's Internet publishing is temporary and unreliable; there is no way to make a believable promise that a document will be available for 50 years or longer. We are investigating the organizational arrangements that will make such promises possible and credible.

transcopy.e36.txt

Date of this version: 95.4.9

Transcopyright: Pre-Permission for Virtual Republishing
Theodor Holm Nelson.

Preliminary Short Version.
APPROVED BY LEGAL COUNSEL.
Note that a repository location has not been established
but should be found soon.
Until then, to cite this article, contact the author.

Note that CERTAIN KEY WORDINGS ARE NOT FINALIZED.
Note also that ITALICS AND SOME OTHER EMPHASES HAVE BEEN LOST.

Trans(c)1994 Theodor Holm Nelson, admin. open.
Xanadu(R), and the Eternal-Flaming-X symbol(R),
are registered trademarks of Project Xanadu.

Theodor Holm Nelson
Founder of Interactive Media | Founder of Network Publishing
Project Xanadu(R) On-Line Storage and Publishing Agency
3020 Bridgeway #295, Sausalito CA 94965
Tel. 415/ 331-4422
Fax 415/ 332-0136
e-mail ted@xanadu.net
and
Sapporo HyperLab
Doshin Maruyama, 8F
2-20, Odori Nishi 20-chome
Chuo-Ku, Sapporo-shi, Hokkaido 064 JAPAN
Tel. (from the States) 011 81 11 621-1133
Fax (from the States) 011 81 11 621-1170
e-mail ted@xanadu.net

SUMMARY

The on-line copyright problem may be resolvable by a simple, sweeping perm

A wording and an abbreviation for this permission system are proposed here

INTRODUCTION

Someone who asks, "How do we keep people from stealing food?" is asking yo

That is how I see the issue of copyright in digital media. Those who mere

Today's central controversy seems to be the question of how to manage copy

The standard question has been, "How do we prevent infringement?" If we r

This reformulates an idea which occurred to the author in the fall of 1960

In this paper, we now separate that copyright method from its software his

transcopy.s16.txt

Date of this version: 95.4.9

Transcopyright: Pre-Permission for Virtual Republishing
Theodor Holm Nelson.

Preliminary Short Version.
APPROVED BY LEGAL COUNSEL.
Note that a repository location has not been established
but should be found soon.
Until then, to cite this article, contact the author.

Note that CERTAIN KEY WORDINGS ARE NOT FINALIZED.
Note also that ITALICS AND SOME OTHER EMPHASES HAVE BEEN LOST.

Xana(c)1994 Theodor Holm Nelson, admin. open.
Xanadu(R), and the Eternal-Flaming-X symbol(R),
are registered trademarks of Project Xanadu.

Theodor Holm Nelson
Founder of Interactive Media (Founder of Newsweek Publishing
Project Xanadu(R) On-line Storage and Publishing Agency
3020 Bridgeway #295, Sausalito CA 94965
Tel. 415/ 331-4422
Fax 415/ 332-0136
e-mail ted@xanadu.net
and
Sapporo Hyperlab
Doshin Maruyama, PP
2-20, Odori Nishi 20-chome
Chuo-Ku, Sapporo-shi, Hokkaido 064 JAPAN
Tel. (from the States) 011 81 11 621-1113
Fax (from the States) 011 81 11 621-1170
e-mail ted@xanadu.net

SUMMARY

The on-line copyright problem may be resolvable by a simple, sweeping perm
A wording and an abbreviation for this permission system are proposed here

INTRODUCTION

Someone who asks, "How do we keep people from stealing food?" is asking yo
That is how I see the issue of copyright in digital media. Those who mere
Today's central controversy seems to be the question of how to manage copy
The standard question has been, "How do we prevent infringement?" If we r
This reformulates an idea which occurred to the author in the fall of 1960
In this paper, we now separate that copyright method from its software hi

THE OBJECTIVE

From the beginning, the objective of Project Xanadu was to facilitate a ne

But this would not mean simply electronic availability of closed documents

The idea was to preserve integrity, copyright and royalty for digital mate

The scheme is simple: under this arrangement, everyone is free to republis

This method provides each receiving party with a properly-owned, independe

A PERMISSION DOCTRINE

What is proposed here turns out actually to be, at its center, a permissic

In this proposed permission doctrine, the copyright holder gives permissic

LONG FORM PERMISSION

To allow this, the copyright holder must therefore make a statement like t

"Permission is granted to all parties in the universe to re-publish the by

This may be adequate to achieve the desired result, as we will consider be

IRONICALLY SIMPLE

This may seem over-simple, when mighty technical and legislative solutions

The new ground becomes the issue of CONTEXT RELINQUISHMENT-- the publisher

This is a tradeoff. We expect publishers-- especially small, non-traditic

IMPLIED ASPECTS

Implied in this method are certain other aspects of implementation. For i

SUPPORTING SOFTWARE ASPECTS

This permission doctrine has certain minimal requirements for supporting s

Server-side software: The server program should have the ability to sell n

Client-side software: Certain detailed features of an unusual nature are

1. Viewer program with purchase methods. Any recipient must have a viewe

2. Editing program which retains identities of portions. Any republisher

3. Stripping program. After an author has finished creating a document w

OWNERSHIP OF EACH FINAL COPY

The recipient who purchases the suggested materials owns a final copy, lik

COMPATIBILITY, INTEROPERABILITY

Note that any party may offer transcopyright; and note that different vari

THE OBJECTIVE

From the beginning, the objective of Project Xanadu was to facilitate a ne

But this would not mean simply electronic availability of closed documents

The idea was to preserve integrity, copyright and royalty for digital mate

The scheme is simple: under this arrangement, everyone is free to republis

This method provides each receiving party with a properly-owned, independe

A PERMISSION DOCTRINE

What is proposed here turns out actually to be, at its center, a permissio

In this proposed permission doctrine, the copyright holder gives permissio

LONG FORM PERMISSION

To allow this, the copyright holder must therefore make a statement like t

"permission is granted to all parties in the universe to re-publish the b

This may be adequate to achieve the desired result, as we will consider b

IRONICALLY SIMPLE

This may seem over-simple, when mighty technical and legislative solutions

The new ground becomes the issue of CONTEXT RELINQUISHMENT-- the publisher

This is a tradeoff. We expect publishers-- especially small, non-traditio

IMPLIED ASPECTS

Implied in this method are certain other aspects of implementation. For i

SUPPORTING SOFTWARE ASPECTS:

This permission doctrine has certain minimal requirements for supporting s

Server-side software: The server program should have the ability to mail a

Client-side software: Certain detailed features of an unusual nature are

1. Viewer program with purchase methods. Any recipient must have a view

2. Editing program which retains identities of portions. Any republisher

3. Stripping program. After an author has finished creating a document w

OWNERSHIP OF EACH FINAL COPY

The recipient who purchases the suggested materials owns a final copy, lik

COMPATIBILITY, INTEROPERABILITY

Note that any party may offer transcopyright; and note that different vari

THE PROPOSED FORMAT AND ITS LEGAL STATUS

I propound for this permission system an extremely simple presentational f

Trans(c)(year)(name)

For example,

Trans(c)1995 Irving Snerd

This is a shorthand form of the long-form permission stated earlier, so th

Some distribution agency or availability scheme should also be specified k

Trans(c)1995 Irving Snerd; administered Dun & Bradstreet.

or

Ziff-Davis Trans(c)1995 Irving Snerd.

Either of these notifications tells all parties, in condensed form, (1) th

It will be noted that the traditional copyright notice

(c)1995 Irving Snerd

is FULLY CONTAINED within the transcopyright notice, and so is still prese

However, note also that this traditional copyright notice no longer has it

By what authority is this a shorthand of the longer version? I hereby DEC

VARIATIONS NOT COVERED;
DIFFERENT JURISDICTIONS

This arrangement is for on-line screen viewing and keepage of digital data

A number of issues are not handled by this arrangement. These include byt

Who settles these details? This is within the scope of variation of the i

ENFORCEMENT AND BENEFITS

This system is not automatically enforced, but neither is any other feasik

---

ACKNOWLEDGMENT

My especial thanks to Robert W. Fiddler, Esq., of Great Neck, N.Y. for ove

BIBLIOGRAPHY

1. Nelson, Theodor Holm, Literary Machines. Availabile from Mindful Pre

2. Nelson, Theodor Holm, "The Art of Connection: Literature Unified by

THE PROPOSED FORMAT AND ITS LEGAL STATUS

I propound for this permission system an extremely simple presentational f

Trans(c)(year)(name)

For example,

Trans(c)1995 Irving Snerd

This is a shorthand form of the long-form permission stated earlier, so th

Some distribution agency or availability scheme should also be specified i

Trans(c)1995 Irving Snerd; administered Dun & Bradstreet.

or

Ziff-Davis Trans(c)1995 Irving Snerd.

Either of these notifications tells all parties, in condensed form, (1) th

It will be noted that the traditional copyright notice

(c)1995 Irving Snerd

is FULLY CONTAINED within the transcopyright notice, and so is still prese

However, note also that this traditional copyright notice no longer has it

By what authority is this a shorthand of the longer version? I hereby DEC

VARIATIONS NOT COVERED;
DIFFERENT JURISDICTIONS

This arrangement is for on-line screen viewing and keepage of digital date

A number of issues are not handled by this arrangement. These include hyi

Who settles these details? This is within the scope of variation of the i

ENFORCEMENT AND BENEFITS

This system is not automatically enforced, but neither is any other feasi

---

ACKNOWLEDGMENT

My especial thanks to Robert W. Fiddler, Esq., of Great Neck, N.Y., for ov

BIBLIOGRAPHY

1. Nelson, Theodor Holm, Literary Machines. Available from Mindful Pre

2. Nelson, Theodor Holm, "The Art of Connection: Literature Unified by

Basics & TRANSPUBLISHING, TRANSQUOTATION AND TRANSCOPYRIGHT
~~to "Transpublishing & transcopyright"~~ to Transpublishing / transcopyright demo
to Ted Nelson home page

TEXT TRANSPUBLISHING FACILITY CENTRAL

Try out your transpublishing page here or publish material offering or using text transcopyright

Welcome to our transpublishing facility for text, where the <TXT SRC> tag acts the way it should.
This is experimental, ~~because~~ at least until we can get the WWW consortium (or certain major Browser makers) to accept <TXT SRC> officially.

Text may be transquoted freely using our source tag. We suggest you also use our special transquotation emblems.

YOU NEED: ① original text material, published ~~not~~ under transcopyright, that you intend to quote.
② the document where you intend to quote it.

INCLUDE THIS SOURCE STRING:

< ~~left quote~~ transquote mark url < > < > right transquote mark
range
specification of quote.

Now: try out your transpublishing page ~~here~~ url: SHOW IT

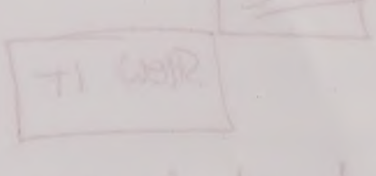
~~Publish~~
Make available text under transcopyright by send the URL by email SEND
Give us the URL of a transquoting page you've published SH

TEXT TRANSPORTING CENTERS

Give us the URL of a transportation page where people publish [illegible] [illegible]

Make available text under transportation [illegible] so we can [illegible]

Now: try out your transportation page here:



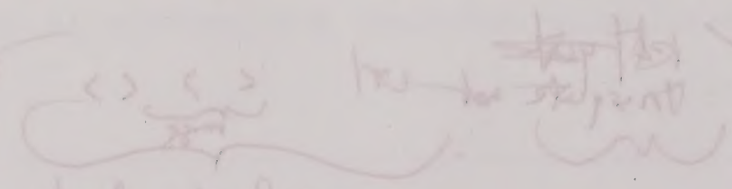
Before go our transportation [illegible] for texts where the CAT [illegible] with them it [illegible].
This is [illegible] at least until we can get the NWW community (or [illegible] [illegible]) to accept CAT URLs officially.

Text may be transported [illegible] URL, our source tag.
We suggest you also use our special transportation alias:

YOU NEED: ① our proxy for [illegible] public [illegible] under [illegible] that you intend to [illegible].

② the address where you [illegible] to push it:

INCLUDE this source string:



Methods for implementing transclusion of text into HTML pages

Copyright (c) 13/06/1996 Andrew Pam of Xanadu Australia
Second draft 28/06/1996 by Andrew Pam
Third draft 24/08/96 by Andrew Pam
This is a draft document and not to be regarded as final.

Introduction

An important requirement of Xanalogical systems is the ability to transclude, or virtually include, portions of one or more documents into another. This enables composite documents to be constructed where each reader obtains the pieces from the original publisher. HTML already permits images in formats including X bitmaps, GIF and JFIF (JPEG) to be transcluded with the <IMG> tag and other document types with the <EMBED> tag but unfortunately does not support the transclusion of text. This document examines methods by which support for text transclusion can be implemented with currently available technologies on the WWW.

1. Java applet

The applet would take the URL and range to transclude as parameters and would attempt to retrieve the requested text and display it within the rectangular space reserved for use by the applet.

Note that the applet can be used because it can cause a URL to be loaded into a window, but can not access data from a URL without visibly loading the entire page.

HTML:

<<TEXT>>

PROS:

Allows the continuation of the page to continue loading and rendering while text transclusions are being retrieved. Doesn't require any changes to the server. Supported by many browsers with more to follow. The code can automatically and transparently be downloaded when required.

CONS:

Not supported by other browsers that don't have Java. Doesn't use a distinct <TEXT> HTML tag. Tags, hotlists and other fonts are probably not implemented yet unless we do it from scratch. Transcluded text must appear within the reserved rectangular applet window and probably can't be richly formatted as with HTML.

2. Netscape plug-in

Documents containing text transclusions would have to be given a particular file extension (for example *.thtml) and MIME-type (for example text/thtml). When such a document is requested the plug-in would be invoked in the background and would parse the incoming HTML stream looking for <TEXT> tags and forwarding the rest, and the result of retrieving the transclusions, to the invoking Netscape window.

HTML:

Methods for implementing transclusion of text into HTML pages

Copyright (c) 13/06/1996 Andrew Pam of Xanadu Australia
Second draft 28/06/1996 by Andrew Pam
Third draft 04/08/96 by Andrew Pam
This is a draft document and not to be regarded as final.

Introduction

An important requirement of Xanalogical systems is the ability to transclude, or virtually include, portions of one or more documents into another. This enables composite documents to be constructed where each reader obtains the pieces from the original publisher. HTML already permits images in formats including X bitmaps, GIF and JFIF (JPEG) to be transcluded with the <IMG> tag and other document types with the <EMBED> tag but unfortunately does not support the transclusion of text. This document examines methods by which support for text transclusion can be implemented with currently available technologies on the WWW.

1. Java applet

The applet would take the URL and range to transclude as parameters and would attempt to retrieve the requested text and display it within the rectangular space reserved for use by the applet.

Note: JavaScript cannot be used because it can cause a URL to be loaded into a window, but can not access data from a URL without visibly loading the entire page.

HTML:
<APPLET>
PROS:
Allows the remainder of the page to continue loading and rendering while text transclusions are being retrieved. Doesn't require any changes to the server. Supported by many browsers with more to follow. The code can automatically and transparently be downloaded when required.
CONS:
Not supported by older browsers that don't have Java. Doesn't use a distinct <TEXT> HTML tag. Japanese and other fonts are probably not implemented yet unless we do it from scratch. Transcluded text must appear within the reserved rectangular applet window and probably can't be richly formatted as with HTML.

2. Netscape plug-in

Documents containing text transclusions would have to be given a particular file extension (for example *.thtml) and MIME-type (for example text/thtml). When such a document is requested the plug-in would be invoked in the background and would parse the incoming HTML stream looking for <TEXT> tags and forwarding the rest, and the result of retrieving the transclusions, to the invoking Netscape window.

HTML:

Methods for implementing transclusion of text into HTML pages

Copyright (c) 13/06/1996 Andrew Pam of Canada Australia
Second draft 28/06/1996 by Andrew Pam
Third draft 04/08/96 by Andrew Pam
This is a draft document and not to be regarded as final.

Introduction

An important requirement of Xanalogical systems is the ability to transclude, or virtually include, portions of one or more documents into another. This enables composite documents to be constructed where each reader obtains the pieces from the original publisher. HTML already permits images in formats including X bitmaps, GIF and JPF (JPEG) to be transcluded with the <IMG> tag and other document types with the <EMBED> tag but unfortunately does not support the transclusion of text. This document examines methods by which support for text transclusion can be implemented with currently available technologies on the WWW.

1. Java applet

The applet would take the URL and range to transclude as parameters and would attempt to retrieve the requested text and display it within the rectangular space reserved for use by the applet.

Note: JavaScript cannot be used because it can cause a URL to be loaded into a window, but can not access data from a URL without visibly loading the entire page.

HTML:

<APPLET>

PROS:

Allows the remainder of the page to continue loading and rendering while text transclusions are being retrieved. Doesn't require any changes to the server. Supported by many browsers with more to follow. The code can automatically and transparently be downloaded when required.

CONS:

Not supported by older browsers that don't have Java. Doesn't use a distinct <TEXT> HTML tag. Japanese and other fonts are probably not implemented yet unless we do it from scratch. Transcluded text must appear within the reserved rectangular applet window and probably can't be richly formatted as with HTML.

2. Netscape plug-in

Documents containing text transclusions would have to be given a particular file extension (for example *.thtml) and MIME-type (for example text/thtml). When such a document is requested the plug-in would be invoked in the background and would parse the incoming HTML stream looking for <TEXT> tags and forwarding the text, and the result of retrieving the transclusions, to the invoking Netscape window.

HTML:

<TEXT>

PROS:

Supports full HTML markup including Japanese and other foreign fonts supported by the browser.

CONS:

Not supported by the many browsers that don't implement Netscape plug-ins. Requires the user to download the plug-in before they can correctly view pages containing text transclusions. Requires server administrators to configure a new MIME-type and publishers to name files containing text transclusions with a different file extension. Finally, the necessary features are not yet implemented by Netscape!

3. Server Side Include with a CGI script

Documents would invoke a CGI script which would attempt to retrieve the URL and range specified as parameters to the script.

HTML:

<!-- #include --> or <!-- #exec -->

PROS:

Supports full HTML markup.

CONS:

Not supported by servers that don't implement server side included CGI scripts. Doesn't use a distinct <TEXT> HTML tag. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

4. Parser/filter CGI script

The script would retrieve each HTML page at the URL specified as its parameter and parse it, looking for <TEXT> tags and attempting to retrive the requested URL and range, then inserting the retrieved text into the HTML page as it is output. The script should also parse hyperlinks (<A HREF> tags) and change the destination to lead back to the script itself with the original link destination as a script parameter, so that the script will continue to be invoked to parse all HTML pages retrieved even when links are followed.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known servers and browsers.

CONS:

Requires that documents containing text transclusions be accessed via the CGI script. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

5. Browser implementation

The browser would directly recognise and interpret <TEXT> tags in HTML to request the specified range and URL and insert it inline.

HTML:

<TEXT>

<TEXT>

PROS:

Supports full HTML markup including Japanese and other foreign fonts supported by the browser.

CONS:

Not supported by the many browsers that don't implement Netscape plug-ins. Requires the user to download the plug-in before they can correctly view pages containing text transclusions. Requires server administrators to configure a new MIME-type and publishers to name files containing text transclusions with a different file extension. Finally, the necessary features are not yet implemented by Netscape!

3. Server Side Include with a CGI script

Documents would invoke a CGI script which would attempt to retrieve the URL and range specified as parameters to the script.

HTML:

<!-- #include --> or <!-- #exec -->

PROS:

Supports full HTML markup.

CONS:

Not supported by servers that don't implement server side included CGI scripts. Doesn't use a distinct <TEXT> HTML tag. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

4. Parser/filter CGI script

The script would retrieve each HTML page at the URL specified as its parameter and parse it, looking for <TEXT> tags and attempting to retrive the requested URL and range, then inserting the retrieved text into the HTML page as it is output. The script should also parse hyperlinks (<A HREF> tags) and change the destination to lead back to the script itself with the original link destination as a script parameter, so that the script will continue to be invoked to parse all HTML pages retrieved even when links are followed.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known servers and browsers.

CONS:

Requires that documents containing text transclusions be accessed via the CGI script. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

5. Browser implementation

The browser would directly recognize and interpret <TEXT> tags in HTML to request the specified range and URL and insert it inline.

HTML:

<TEXT>

PROS:

Probably the most efficient solution.

CONS:

Doesn't support users of other browsers.

6. Server gateway

The server would parse all HTML files and request the necessary transcluded material while serving each document. Mr. Yousuke Igarashi <yousuke@crew.sfc.keio.ac.jp> suggested making this a proxy module for a web server such as Apache, which would allow users of other servers to set a transclusion supporting server as their proxy.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known browsers.

CONS:

Doesn't support users of other servers. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

Because all of these methods have their strengths and weaknesses we will probably want to implement more than one. I believe we decided to start with method 4 (CGI script) and probably methods 5 and 6 later.

I propose that the new HTML tag should be something like this:

<TEXT SRC=[URL] {(PLAIN|RANGE)={[start]},{[end]}} {WIDTH=[X] HEIGHT=[Y]}>

Where braces {} enclose optional elements, brackets [] enclose variable parameters and parentheses () contain mutually exclusive alternatives separated by vertical bars |.

Parameters:

SRC=[URL]

Mandatory. Specifies the source document from which text is to be transcluded. [URL] must be the URL of a text document of some kind, HTML or otherwise.

(PLAIN|RANGE)={[start]},{[end]}

Optional. If this parameter is omitted, the source document will be transcluded in its entirity. [start] and [end] must be byte offsets within the file. If PLAIN is used, only the text of the source document is parsed; all tags are omitted both in determining the specified offsets and in the transclusion. If RANGE is used, the source document is transcluded verbatim. It is probably an error for both [start] and [end] to be omitted. If either or both are out of range any in range portion selected should probably still be delivered.

WIDTH=[X] HEIGHT=[Y]

Optional. [X] and [Y] are in pixels and would allow the browser to reserve a rectangular space to present the transcluded text rather than having to wait for it to arrive before continuing the layout of the page, in exactly the same fashion as the <IMG> tag is typically implemented. If the transcluded text will not fit in the reserved space scroll bars could be displayed. It probably doesn't make sense to implement support for these parameters in the

PROS:

Probably the most efficient solution.

CONS:

Doesn't support users of other browsers.

6. Server gateway

The server would parse all HTML files and request the necessary transcluded material while serving each document. Mr. Yousuke Igarashi <yousuke@crew.sfc.keio.ac.jp> suggested making this a proxy module for a web server such as Apache, which would allow users of other servers to set a transclusion supporting server as their proxy.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known browsers.

CONS:

Doesn't support users of other servers. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

Because all of these methods have their strengths and weaknesses we will probably want to implement more than one. I believe we decided to start with method 4 (CGI script) and probably methods 5 and 6 later.

I propose that the new HTML tag should be something like this:

<TEXT SRC=[URL] {(PLAINRANGE)=([start]),([end])} {(WIDTH=[X] HEIGHT=[Y])}>

Where braces {} enclose optional elements, brackets [] enclose variable parameters and parentheses () contain mutually exclusive alternatives separated by vertical bars |.

Parameters:

SRC=[URL]

Mandatory. Specifies the source document from which text is to be transcluded. [URL] must be the URL of a text document of some kind, HTML or otherwise.

(PLAINRANGE)=([start]),([end])

Optional. If this parameter is omitted, the source document will be transcluded in its entirety. [start] and [end] must be byte offsets within the file. If PLAIN is used, only the text of the source document is parsed; all tags are omitted both in determining the specified offsets and in the transclusion. If RANGE is used, the source document is transcluded verbatim. It is probably an error for both [start] and [end] to be omitted. If either or both are out of range any in range portion selected should probably still be delivered.

WIDTH=[X] HEIGHT=[Y]

Optional. [X] and [Y] are in pixels and would allow the browser to reserve a rectangular space to present the transcluded text rather than having to wait for it to arrive before continuing the layout of the page, in exactly the same fashion as the <IMG> tag is typically implemented. If the transcluded text will not fit in the reserved space scroll bars could be displayed. It probably doesn't make sense to implement support for these parameters in the

server-side implementations (3, 4 and 6).

The intention is to have a facility in authoring programs that permits the author to create transclusions by indicating an insertion point, viewing the document from which they wish to transclude, and marking the region to be transcluded, much in the manner of a traditional "cut and paste" operation except that what is actually pasted is the reference to the transcluded portion rhather than the literal text.

Initially, this could be a small editing program purely for adding transclusions to existing documents. It has also been suggested that people might wish to add transclusions by hand, in which case it might be desirable to have other ways of specifying the start and end of the range besides just the byte offsets, which are inconvenient to determine by hand.

Possible extensions to [start] and [end] values (for discussion):

1. HTML target anchors <A NAME="target">, indicated by prefixing the target name with a hash mark. This should reference the start of an anchor. Byte offsets from this position could be permitted by appending a plus or minus and the number of bytes. Examples:

   ```
   RANGE=#start,#end
   RANGE=#start+5,#end-1
   ```

2. Paragraphs (<P> in HTML), indicated by the letter P and the paragraph number counting from the beginning of the document. Sentences and words could also be supported similarly to paragraphs, but at additional computational expense. Example:

   ```
   RANGE=P5,P9-3
   ```

3. Offsets from pattern matches, suggested by Paul Haeberli <paul@sgi.com> in his "Merge" script. This could be signified by enclosing the pattern with slashes, single or double quotes as delimiters which may not appear within the pattern. Examples:

   ```
   RANGE='<img src',"</center>"-15
   RANGE=105,/day./+6
   ```

Comments welcome!

server-side implementations (3, 4 and 6).

The intention is to have a facility in authoring programs that permits the author to create transclusions by indicating an insertion point, viewing the document from which they wish to transclude, and marking the region to be transcluded, much in the manner of a traditional "cut and paste" operation except that what is actually pasted is the reference to the transcluded portion rather than the literal text.

Initially, this could be a small editing program purely for adding transclusions to existing documents. It has also been suggested that people might wish to add transclusions by hand, in which case it might be desirable to have other ways of specifying the start and end of the range besides just the byte offsets, which are inconvenient to determine by hand.

Possible extensions to [start] and [end] values (for discussion):

1. HTML target anchors <A NAME="target">, indicated by prefixing the target name with a hash mark. This should reference the start of an anchor. Byte offsets from this position could be permitted by appending a plus or minus and the number of bytes. Examples:

```
RANGE=#start,#end
RANGE=#start+5,#end-1
```

2. Paragraphs (<P> in HTML), indicated by the letter P and the paragraph number counting from the beginning of the document. Sentences and words could also be supported similarly to paragraphs, but at additional computational expense. Example:

```
RANGE=p5,p9-3
```

3. Offsets from pattern matches, suggested by Paul Hasbrell <paul@sgi.com> in his "Merge" script. This could be signified by enclosing the pattern with slashes, single or double quotes as delimiters which may not appear within the pattern. Examples:

```
RANGE=<'<img src\ ".*\/contents\"-15
RANGE=105,\day.\*6
```

Comments welcome!

Methods for implementing transclusion of text into HTML pages

Copyright (c) 13/06/1996 Andrew Pam of Xanadu Australia
Second draft 28/06/1996 by Andrew Pam
Third draft 04/08/96 by Andrew Pam
This is a draft document and not to be regarded as final.

Introduction

An important requirement of Xanalogical systems is the ability to transclude, or virtually include, portions of one or more documents into another. This enables composite documents to be constructed where each reader obtains the pieces from the original publisher. HTML already permits images in formats including X bitmaps, GIF and JFIF (JPEG) to be transcluded with the <IMG> tag and other document types with the <EMBED> tag but unfortunately does not support the transclusion of text. This document examines methods by which support for text transclusion can be implemented with currently available technologies on the WWW.

1. Java applet

The applet would take the URL and range to transclude as parameters and would attempt to retrieve the requested text and display it within the rectangular space reserved for use by the applet.

Note: JavaScript cannot be used because it can cause a URL to be loaded into a window, but can not access data from a URL without visibly loading the entire page.

HTML:
<APPLET>

PROS:
Allows the remainder of the page to continue loading and rendering while text transclusions are being retrieved. Doesn't require any changes to the server. Supported by many browsers with more to follow. The code can automatically and transparently be downloaded when required.

CONS:
Not supported by older browsers that don't have Java. Doesn't use a distinct <TEXT> HTML tag. Japanese and other fonts are probably not implemented yet unless we do it from scratch. Transcluded text must appear within the reserved rectangular applet window and probably can't be richly formatted as with HTML.

2. Netscape plug-in

Documents containing text transclusions would have to be given a particular file extension (for example *.thtml) and MIME-type (for example text/thtml). When such a document is requested the plug-in would be invoked in the background and would parse the incoming HTML stream looking for <TEXT> tags and forwarding the rest, and the result of retrieving the transclusions, to the invoking Netscape window.

HTML:

Methods for implementing transclusion of text into HTML pages

Copyright (c) 13/06/1996 Andrew Pam of Xanadu Australia
Second draft 26/06/1996 by Andrew Pam
Third draft 04/08/96 by Andrew Pam
This is a draft document and not to be regarded as final.

Introduction

An important requirement of Xanalogical systems is the ability to transclude, or virtually include, portions of one or more documents into another. This enables composite documents to be constructed where each reader obtains the pieces from the original publisher. HTML already permits images in formats including X bitmaps, GIF and JPF (JPEG) to be transcluded with the <IMG> tag and other document types with the <EMBED> tag but unfortunately does not support the transclusion of text. This document examines methods by which support for text transclusion can be implemented with currently available technologies on the WWW.

1. Java applet

The applet would take the URL and range to transclude as parameters and would attempt to retrieve the requested text and display it within the rectangular space reserved for use by the applet.

Note: JavaScript cannot be used because it can cause a URL to be loaded into a window, but can not access data from a URL without visibly loading the entire page.

HTML:

<APPLET>

PROS:

Allows the remainder of the page to continue loading and rendering while text transclusions are being retrieved. Doesn't require any changes to the server. Supported by many browsers with more to follow. The code can automatically and transparently be downloaded when required.

CONS:

Not supported by older browsers that don't have Java. Doesn't use a distinct <TEXT> HTML tag. Japanese and other fonts are probably not implemented yet unless we do it from scratch. Transcluded text must appear within the reserved rectangular applet window and probably can't be richly formatted as with HTML.

2. Netscape plug-in

Documents containing text transclusions would have to be given a particular file extension (for example *.thtml) and MIME-type (for example text/thtml). When such a document is requested the plug-in would be invoked in the background and would parse the incoming HTML stream looking for <TEXT> tags and forwarding the rest, and the result of retrieving the transclusions, to the invoking Netscape window.

HTML:

<TEXT>

PROS:

Supports full HTML markup including Japanese and other foreign fonts supported by the browser.

CONS:

Not supported by the many browsers that don't implement Netscape plug-ins. Requires the user to download the plug-in before they can correctly view pages containing text transclusions. Requires server administrators to configure a new MIME-type and publishers to name files containing text transclusions with a different file extension. Finally, the necessary features are not yet implemented by Netscape!

3. Server Side Include with a CGI script

Documents would invoke a CGI script which would attempt to retrieve the URL and range specified as parameters to the script.

HTML:

<!-- #include --> or <!-- #exec -->

PROS:

Supports full HTML markup.

CONS:

Not supported by servers that don't implement server side included CGI scripts. Doesn't use a distinct <TEXT> HTML tag. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

4. Parser/filter CGI script

The script would retrieve each HTML page at the URL specified as its parameter and parse it, looking for <TEXT> tags and attempting to retrive the requested URL and range, then inserting the retrieved text into the HTML page as it is output. The script should also parse hyperlinks (<A HREF> tags) and change the destination to lead back to the script itself with the original link destination as a script parameter, so that the script will continue to be invoked to parse all HTML pages retrieved even when links are followed.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known servers and browsers.

CONS:

Requires that documents containing text transclusions be accessed via the CGI script. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

5. Browser implementation

The browser would directly recognise and interpret <TEXT> tags in HTML to request the specified range and URL and insert it inline.

HTML:

<TEXT>

<TEXT>

PROS:

Supports full HTML markup including Japanese and other foreign fonts supported by the browser.

CONS:

Not supported by the many browsers that don't implement Netscape plug-ins. Requires the user to download the plug-in before they can correctly view pages containing text transclusions. Requires server administrators to configure a new MIME-type and publishers to name files containing text transclusions with a different file extension. Finally, the necessary features are not yet implemented by Netscape!

3. Server Side Include with a CGI script

Documents would invoke a CGI script which would attempt to retrieve the URL and range specified as parameters to the script.

HTML:

<!– #include –> or <!– #exec –>

PROS:

Supports full HTML markup.

CONS:

Not supported by servers that don't implement server side included CGI scripts. Doesn't use a distinct <TEXT> HTML tag. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

4. Parser/filter CGI script

The script would retrieve each HTML page at the URL specified as its parameter and parse it, looking for <TEXT> tags and attempting to retrieve the referenced URL and range, then inserting the retrieved text into the HTML page as it is output. The script should also parse hyperlinks (<A HREF> tags) and change the destination to lead back to the script itself with the original link destination as a script parameter, so that the script will continue to be invoked to parse all HTML pages retrieved even when links are followed.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known servers and browsers.

CONS:

Requires that documents containing text transclusions be accessed via the CGI script. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

5. Browser implementation

The browser would directly recognize and interpret <TEXT> tags in HTML to request the specified range and URL and insert it inline.

HTML:

<TEXT>

PROS:

Probably the most efficient solution.

CONS:

Doesn't support users of other browsers.

6. Server gateway

The server would parse all HTML files and request the necessary transcluded material while serving each document. Mr. Yousuke Igarashi <yousuke@crew.sfc.keio.ac.jp> suggested making this a proxy module for a web server such as Apache, which would allow users of other servers to set a transclusion supporting server as their proxy.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all known browsers.

CONS:

Doesn't support users of other servers. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

Because all of these methods have their strengths and weaknesses we will probably want to implement more than one. I believe we decided to start with method 4 (CGI script) and probably methods 5 and 6 later.

I propose that the new HTML tag should be something like this:

<TEXT SRC=[URL] {(PLAIN|RANGE)={[start]},{[end]}} {WIDTH=[X] HEIGHT=[Y]}>

Where braces {} enclose optional elements, brackets [] enclose variable parameters and parentheses () contain mutually exclusive alternatives separated by vertical bars |.

Parameters:

SRC=[URL]

Mandatory. Specifies the source document from which text is to be transcluded. [URL] must be the URL of a text document of some kind, HTML or otherwise.

(PLAIN|RANGE)={[start]},{[end]}

Optional. If this parameter is omitted, the source document will be transcluded in its entirity. [start] and [end] must be byte offsets within the file. If PLAIN is used, only the text of the source document is parsed; all tags are omitted both in determining the specified offsets and in the transclusion. If RANGE is used, the source document is transcluded verbatim. It is probably an error for both [start] and [end] to be omitted. If either or both are out of range any in range portion selected should probably still be delivered.

WIDTH=[X] HEIGHT=[Y]

Optional. [X] and [Y] are in pixels and would allow the browser to reserve a rectangular space to present the transcluded text rather than having to wait for it to arrive before continuing the layout of the page, in exactly the same fashion as the <IMG> tag is typically implemented. If the transcluded text will not fit in the reserved space scroll bars could be displayed. It probably doesn't make sense to implement support for these parameters in the

PROS:

Probably the most efficient solution.

CONS:

Doesn't support users of other browsers.

6. Server gateway

The server would parse all HTML files and request the necessary transcluded material while serving each document. Mr. Yousuke Igarashi <yousuke@crew.sfc.keio.ac.jp> suggested making this a proxy module for a web server such as Apache, which would allow users of other servers to set a transclusion supporting server as their proxy.

HTML:

<TEXT>

PROS:

Supports full HTML markup. Should work with all browse browsers.

CONS:

Doesn't support users of other servers. Transcluded text is retrieved by the server on behalf of the user, rather than directly by the user.

Because all of these methods have their strengths and weaknesses we will probably want to implement more than one. I believe we decided to start with method 4 (CGI script) and probably methods 5 and 6 later.

I propose that the new HTML tag should be something like this:

<TEXT SRC=[URL] {PLAINRANGE}={[start]}{[end]} {WIDTH=[X] HEIGHT=[Y]}>

Where braces {} enclose optional elements; brackets [] enclose variable parameters and parentheses () contain mutually exclusive alternatives separated by vertical bars |.

Parameters:

SRC=[URL]

Mandatory. Specifies the source document from which text is to be transcluded. [URL] must be the URL of a text document of some kind, HTML or otherwise.

{PLAINRANGE}={[start]}:{[end]}

Optional. If this parameter is omitted, the source document will be transcluded in its entirity. [start] and [end] must be byte offsets within the file. If PLAIN is used, only the text of the source document is parsed; all tags are omitted both in determining the specified offsets and in the transclusion. If RANGE is used, the source document is transcluded verbatim. It is probably an error for both [start] and [end] to be omitted. If either or both are out of range any in range portion selected should probably still be delivered.

WIDTH=[X] HEIGHT=[Y]

Optional. [X] and [Y] are in pixels and would allow the browser to reserve a rectangular space to present the transcluded text rather than having to wait for it to arrive before continuing the layout of the page, in exactly the same fashion as the <IMG> tag is typically implemented. If the transcluded text will not fit in the reserved space scroll bars could be displayed. It probably doesn't make sense to implement support for these parameters in the

server-side implementations (3, 4 and 6).

The intention is to have a facility in authoring programs that permits the author to create transclusions by indicating an insertion point, viewing the document from which they wish to transclude, and marking the region to be transcluded, much in the manner of a traditional "cut and paste" operation except that what is actually pasted is the reference to the transcluded portion rhather than the literal text.

Initially, this could be a small editing program purely for adding transclusions to existing documents. It has also been suggested that people might wish to add transclusions by hand, in which case it might be desirable to have other ways of specifying the start and end of the range besides just the byte offsets, which are inconvenient to determine by hand.

Possible extensions to [start] and [end] values (for discussion):

1. HTML target anchors <A NAME="target">, indicated by prefixing the target name with a hash mark. This should reference the start of an anchor. Byte offsets from this position could be permitted by appending a plus or minus and the number of bytes. Examples:

   ```
   RANGE=#start,#end
   RANGE=#start+5,#end-1
   ```

2. Paragraphs (<P> in HTML), indicated by the letter P and the paragraph number counting from the beginning of the document. Sentences and words could also be supported similarly to paragraphs, but at additional computational expense. Example:

   ```
   RANGE=P5,P9-3
   ```

3. Offsets from pattern matches, suggested by Paul Haeberli <paul@sgi.com> in his "Merge" script. This could be signified by enclosing the pattern with slashes, single or double quotes as delimiters which may not appear within the pattern. Examples:

   ```
   RANGE='<img src',"</center>"-15
   RANGE=105,/day./+6
   ```

Comments welcome!

server-side implementations (3, 4 and 5).

The intention is to have a facility in authoring programs that permits the author to create transclusions by indicating an insertion point, viewing the document from which they wish to transclude, and marking the region to be transcluded, much in the manner of a traditional "cut and paste" operation except that what is actually pasted is the reference to the transcluded portion rhather than the literal text.

Initially, this could be a small editing program purely for adding transclusions to existing documents. It has also been suggested that people might wish to add transclusions by hand, in which case it might be desirable to have other ways of specifying the start and end of the range besides just the byte offsets, which are inconvenient to determine by hand.

Possible extensions to [start] and [end] values (for discussion):

1. HTML target anchors <A NAME="target">, indicated by prefixing the target name with a hash mark. This should reference the start of an anchor. Byte offsets from this position could be permitted by appending a plus or minus and the number of bytes. Examples:

```
RANGE=#start,#end
RANGE=#start+5,#end-1
```

2. Paragraphs (<P> in HTML), indicated by the letter P and the paragraph number counting from the beginning of the document. Sentences and words could also be supported similarly to paragraphs, but at additional computational expense. Example:

```
RANGE=P5,P9-3
```

3. Offsets from pattern matches, suggested by Paul Haeberli <paul@sgi.com> in his "Merge" script. This could be signified by enclosing the pattern with slashes, single or double quotes as delimiters which may not appear within the pattern. Examples:

```
RANGE='<img src',"</center>"-15
RANGE=105,/day./+6
```

Comments welcome!